

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2014

Bc. Martin Střelec

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

Řídicí jednotka měniče pro fotovoltaické panely

Inverter Control Unit for Photovoltaic Panel

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

Zadání diplomové práce

Student: **Bc. Martin Střelec**
Studijní program: N2649 Elektrotechnika
Studijní obor: 2612T015 Elektronika
Téma: **Řídicí jednotka měniče pro fotovoltaické panely**
Inverter Control Unit for Photovoltaic Panel

Zásady pro vypracování:

1. Popište principy přenosu a využití elektrické energie z fotovoltaických panelů s ohledem na možnost připojení k DC/DC měničům.
2. Navrhněte a realizujte řídicí jednotku pro řízení DC/DC měniče fotovoltaického panelu.
3. Vytvořte software pro řídicí jednotku s možností optimálního využití energie panelu.
4. Podle pokynů vedoucího diplomové práce proveďte měření a vyhodnoťte funkčnost algoritmu.

Seznam doporučené odborné literatury:

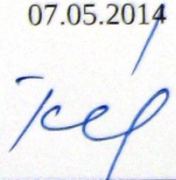
Dle pokynů vedoucího diplomové práce

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Petr Palacký, Ph.D.**

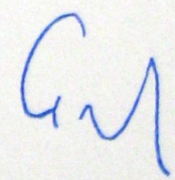
Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Ing. Petr Palacký, Ph.D.
vedoucí katedry





prof. RNDr. Václav Snášel, CSc.
děkan fakulty

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

Čestné prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 7. 5. 2014



.....
Autor

Poděkování

Tímto bych rád poděkoval svému vedoucímu diplomové práce doc. Ing. Petru Palackému, Ph.D. za vedení a odbornou pomoc při řešení této práce a dále doktorandu Ing. Josefu Opluštilovi za rady a pomoc při vytváření programu.

Abstrakt

Cílem práce je vytvořit řídicí část měniče pro fotovoltaické systémy. Diplomová práce se zabývá problematikou ostrovních solárních systémů, ukládáním energie v těchto systémech a maximálním využitím energetického potenciálu fotovoltaických panelů. V úvodu práce jsou popsány způsoby propojení fotovoltaických systémů a druhy měničů využívající tyto systémy. Vytvořený komplet měniče a řídicího systému se přímo zařazuje do energetické soustavy fotovoltaického systému, kde plní funkci měření důležitých veličin, řízení měniče a ochranu fotovoltaických panelů, akumulátorů a zátěže.

Klíčová slova

Fotovoltaický panel, ostrovní systém, DSP, signálový procesor, DC/DC měnič, MPPT

Abstract

The aim is to create a control of the inverter for photovoltaic systems. This diploma thesis deals with grid solar systems, energy storage in these systems and maximum use of the energy potential of photovoltaic panels. In the introduction describes the methods of interconnection of PV systems and inverters using these types of systems. Created complete inverter and control system directly classified in the power system photovoltaic system, which serves as the measurement of relevant variables, drive control and protection of photovoltaic panels, batteries and loads.

Key Words

Photovoltaic panel, island power systems, DSP, signal processor, DC / DC converter, MPPT

Obsah

Obsah.....	5
Úvod	7
Vývoj solárních panelů a současný stav fotovoltaiky	7
<i>Ostrovní solární systémy.....</i>	<i>7</i>
Solární články a panely	8
Solární regulátor	11
Solární akumulátor	13
Měnič pro spotřební zařízení.....	14
Návrh řídicí jednotky DC/DC měniče	17
<i>Zvyšující DC/DC měnič.....</i>	<i>17</i>
<i>Vývojový kit TMS320F28335.....</i>	<i>19</i>
Popis zapojení jednotlivých částí:.....	21
<i>Měření napětí, proudu a teploty.....</i>	<i>23</i>
Měření napětí.....	23
Měření proudu	24
Měření teploty.....	26
<i>Řízení nabíjení akumulátoru</i>	<i>26</i>
Charakteristika U	26
Charakteristika I	26
Charakteristika W	27
Charakteristika UI	27
VRLA akumulátor s Gelovým elektrolytem.....	27
VRLA akumulátor s vázaným elektrolytem AGM.....	28
Teplotní kompenzace	28
Vytvoření programu pro řídicí jednotku	29
<i>Vložení knihoven:.....</i>	<i>29</i>
<i>Definice proměnných:</i>	<i>29</i>
<i>Deklarace funkcí v programu:.....</i>	<i>30</i>
<i>Hlavní smyčka:</i>	<i>30</i>
Inicializace systémových součástí.....	30
Inicializace GPIO	30
Zakázání všech přerušení CPU	30
Inicializace PIE registru na výchozí stav	31
Zakázání přerušení CPU a vynulování všech příznaků	31
Inicializace všech PIE vektorů přerušení na ukazatele ISR.....	31

Přemapování vektoru přerušení.....	31
Inicializace periférií:.....	31
Inicializace PWM	31
Povolení přerušení pro skupinu 1.....	31
Povolení přerušení od ADC v PIE	31
Povolení všech přerušení	32
Inicializace AD převodníku	32
Nekonečná smyčka:.....	32
<i>Přerušení:</i>	32
Manuální nastavení hodnot PWM z LabView	33
Regulační smyčka měniče	34
Trackové řízení	35
Načtení hodnot z AD převodníku	36
Úprava hodnot střídý PWM.....	36
Odeslání hodnot nastavených v LabView na DA převodník	37
Reset měření a potvrzení přerušení	37
Vizualizace rozhraní pomocí LabView.....	37
Měření:	42
<i>Ověření generovaných a měřených hodnot vývojového kitu</i>	<i>42</i>
AD převodník	42
Měření teploty.....	43
<i>Měření s připojeným DC/DC měničem</i>	<i>44</i>
Měnič buzený řídicím systémem v manuálním režimu řízení	44
Měnič buzený řídicím systémem v automatickém režimu	45
Závěr:.....	47
Možnost rozšíření řídicího systému:	47
Literatura:	48
<i>Elektronické zdroje:.....</i>	<i>48</i>
Přílohy.....	I

Úvod

Fotovoltaické systémy jsou jedním z nerychleji se rozšiřujících druhů obnovitelných zdrojů energie. Pro takové zdroje se uvažuje využití minimálně 40let bez úbytku primárního zdroje energie. Primárním cílem využití obnovitelných zdrojů je výroba elektrické energie a tepelné energie. Všechny obnovitelné zdroje jsou závislé na místě určení a tím je omezeno i jejich plošné nasazení. Mezi další obnovitelné zdroje energie patří kromě slunečního záření také geotermální energie, větrná energie, vodní energie, energie mořských proudů, energie biomasy aj.

Diplomová práce spolupracuje s diplomovou prací Bc. Tomáše Chalupky "Měnič pro fotovoltaiku", pro niž vytváří řídicí systém. Takto vytvořená měničová struktura je určena pro zpracování energie přímo vyrobené solárním panelem a její řídicí algoritmus slouží pro maximální využití energetického potenciálu fotovoltaického panelu. Solární panel přímo přeměňuje sluneční záření na elektrickou energii, lze jej tedy pro zjednodušení považovat za zdroj stejnosměrného napětí s energií závislou na intenzitě dopadajícího slunečního záření.

Vývoj solárních panelů a současný stav fotovoltaiky

Fotovoltaický jev je znám od roku 1839, kdy jej objevil francouzský fyzik Alexander Edmond Becquerel. Roku 1883 byl vytvořen první fotovoltaický článek s účinností okolo 1%. Prvním funkčním nasazením fotovoltaiky bylo využití v kosmonautice, kde fotovoltaické články tvořily jediný možný dlouhodobý zdroj energie pro kosmické družice.

Snižující se cena solárních systémů dovoluje jejich rozšiřování do civilního sektoru, kde se připojují na jednotnou distribuční síť, nebo využití na místech bez dostupnosti elektrické energie kde vytváří tzv. ostrovní systémy, např. mobilní zařízení (jachty, karavany), odlehlé oblasti (horské objekty, ropné plošiny).

Fotovoltaika se díky široké možnosti nasazení stala jedním z nejrychleji se rozvíjejících oborů elektroenergetiky. V některých zemích činí meziroční nárůst i více než 30%. Tento nárůst umožnila snižující se cena solárních systémů, zdokonalování technologie výroby a zvyšování účinnosti solárních systémů, motivace ze strany státu a celkový efekt hromadné výroby.

Dnes využívané fotovoltaické články dosahují účinnosti okolo 12-17% při životnosti okolo 30 let. Nejnovější laboratorní výsledky slibují články využívající trojnásobné přechody s účinností až 40%.

Ostrovní solární systémy

Ostrovní solární systémy vycházejí především z nemožnosti připojení objektu k stávající distribuční síti, což může být způsobeno geografickou nedostupností sítě, neekonomickou stránkou připojení nebo mobilností zařízení/objektu. Další možnou aplikací je vytvoření záložního systému pro případ "blackout" tedy výpadku elektrické rozvodné sítě, v takovém případě hovoříme o tzv. hybridním fotovoltaickém systému. Dnešním trendem je také vytváření tzv. "zero - energy building" což jsou objekty maximálně využívající obnovitelných zdrojů s minimálními náklady na provoz a distribuční energie. V takovýchto stavbách se už při projektování uvažuje se solárními elektrickými a

tepelnými systémy, elektrická soustava je navržena nízkonapěťová a s energeticky nenáročnými prvky (12/24V rozvody, LED osvětlení).

Ostrovní solární systém se skládá z několika základních prvků:

Solární články a panely

Solární články se dělí dle technologií výroby, popř. dle nosných podkladových materiálů.

Technologie tlustých vrstev

Fotovoltaický článek je tvořen polovodičovou p-n diodou. Tyto články se vyrábějí z křemíkových plátek, ať už z monokrystalického nebo polykrystalického křemíku. V současné době se touto technologií vyrábí více než 85% solárních článků na trhu.

Technologie tenkých vrstev

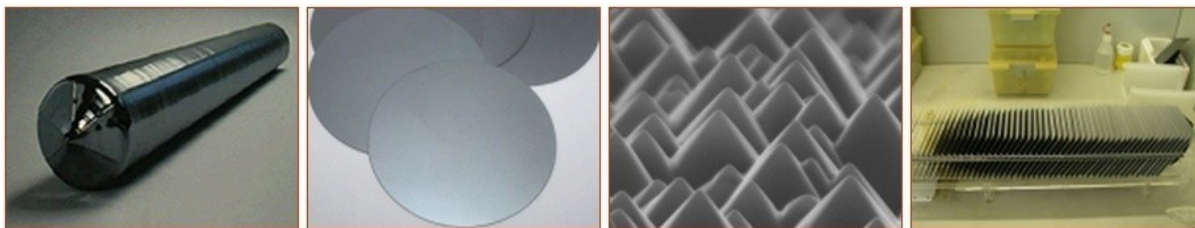
Fotovoltaický článek je tvořen nosnou plochou (například sklem, textilií a podobně), na které jsou napařené velmi tenké vrstvy amorfního nebo mikrokrystalického křemíku. Množství materiálu, použitého pro výrobu tenkovrstvého fotovoltaického článku, je nižší, než u tlustých vrstev, takže články jsou levnější. Nevýhodou současných tenkovrstvých fotovoltaických článků je nižší účinnost a nižší životnost.

Nekřemíkové technologie

Na rozdíl od předešlých dvou se pro konverzi světla na elektrickou energii nepoužívá tradiční P-N polovodičový přechod. Používají se různé organické sloučeniny, polymery a podobně. Tyto technologie jsou většinou ve stadiu výzkumů.

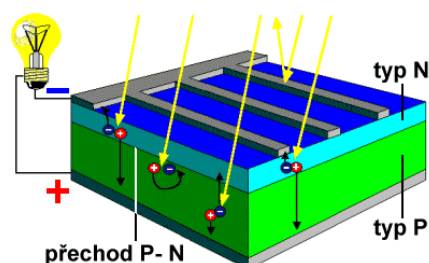
Vzhledem k možnému masovému využití fotovoltaických článků, jejichž výrobní cena by byla podstatně nižší než v současnosti, probíhá také výzkum fotovoltaických článků pracujících s jinými fotocitlivými materiály než je křemík. Jednou z možností jsou vodivé polymery, např. v listopadu 2005 se podařilo výzkumné skupině na University of California v Los Angeles dosáhnout zatím maximální účinnosti 4,4%. [12]

Solární článek je ve svém principu polovodičová dioda ve tvaru tenké desky. Články se vyrábějí stejně jako mikročipy z křemíkových ingotů, které se upraví na obdélníkový / čtvercový průřez (kvůli lepšímu využití plochy budoucích solárních panelů). Ingot se rozřeže na jednotlivé destičky o tloušťce 0,3mm, jež tvoří budoucí základ článků (vrstva P). Dalším postupem je leptání povrchu destiček, čímž se odstraní nedokonalosti po řezu a vytvoří se texturovaný povrch (deska zmatná a méně odráží světlo), což zvyšuje účinnost článku. Poté následuje dopování Fosforem, čímž se vytvoří vrstva N, která má sílu max. 500nm. Následuje antireflexní úprava, která ještě více snižuje odrazivost a zvyšuje mechanickou odolnost. Tato vrstva je tvořena nitridem křemíku a dává článkům typicky modrý nádech. Posledním krokem je vytvoření vodivých kontaktů pro připojení vodičů.



Obr.:1 Křemíkový ingot (1), tenké podkladové destičky (2), detail texturovaného povrchu po leptání (3), příprava na vytvoření vrstvy N (4) [9]

Osvětlením článku vznikne v polovodiči vnitřní fotoelektrický jev a v polovodiči se z krystalové mřížky začnou uvolňovat záporné elektrony. Na přechodu P-N se vytvoří elektrické napětí, které dosahuje u křemíkových článků velikosti zhruba 0,55V.



Obr.:2 Fotovoltaický jev u solárního článku [8]

Proud, který můžeme ze slunečního článku odebírat, je úměrný velikosti ozařované plochy článku. Výkon křemíkových fotovoltaických článků o ploše 1 m^2 je 80 – 100 W, účinnost přeměny světelné energie na energii elektrickou závisí na struktuře materiálu a způsobu výroby článku (laboratorní nebo průmyslová):

KŘEMÍKOVÝ MATERIÁL	laboratorní výr. (%)	průmyslová výr. (%)
monokrystalický	~ 24	14 - 17
polykrystalický	~ 18	13 - 15
amorfni	~ 13	5 - 7

Tab.:1 Porovnání účinnosti solárních článků [8]

Výkon solárního článku se udává v jednotkách W_p (watt peak - špičková hodnota). Výkon závisí na osvětlení článku a úhlu dopadajícího světla. Účinnost solárních článků se určuje při plošném osvětlení 1 kW/m^2 a teplotě článku 25°C , což vyjadřuje ideální sluneční osvětlení.

Jednotlivé solární články se využívají jen velice zřídka, především kvůli jejich malé zatížitelnosti a malému výstupnímu napětí, proto se jednotlivé články sestavují do panelů (modulů). Solární panel je tedy vytvořen sériovým a paralelním řazením jednotlivých solárních článků dle požadovaného napětí a výkonu. Kvůli dosažení vyšší životnosti a jednodušší údržbě jsou panely vytvořeny jako hermeticky uzavřená jednotka zakrytá vysoce průhledným tvrzeným sklem. Takovéto provedení zaručuje ochranu před povětrnostními vlivy po dobu 20-30let. Budoucností ve výrobě pouzder solárních panelů je využívání nanotechnologií především pro dosažení lepších vlastností krycího skla - větší mechanická odolnost při nižší hmotnosti, teplotní a světelná dlouhodobá stálost, mechanická stálost povrchu a odolnost proti zanášení prachem a nečistotami.

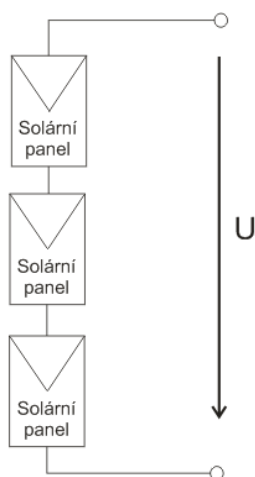
Vyšší účinnosti celého systému se dosahuje použitím zařízení pro sledování slunce (tracker) nebo použitím odrazových zrcadel koncentrujících sluneční záření z větší plochy na menší solární panel. Tyto technická řešení vyžadují složitější zařízení (tracker), nebo častější údržbu (čištění zrcadel), takže jejich nasazení je možné pouze v omezeném množství.

Řazení panelů

Solární panely se standardně vyrábějí z 36 nebo 72 článků s nominálním napětím 12 nebo 24V. Pro získání vyššího nominálního napětí, vyššího odebíraného proudu nebo celkového vyššího výkonu je nutné řadit panely sériově / paralelně. Při tomto řazení je důležité dbát na správné tepelné a proudové dimenzování spojovacích vodičů.

Sériové řazení panelů

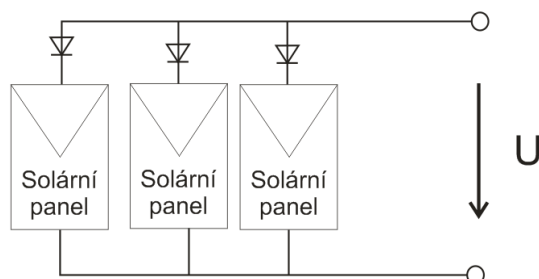
Pro dosažení požadovaného napětí se panely seřadí sériově. Při tomto řazení je nutné dbát na maximální izolační napětí panelů, což bývá typicky 1000V. Sériovým řazením vznikají tzv. stringy (řetězce).



Obr.:3 Sériové řazení solárních panelů

Paralelní řazení panelů

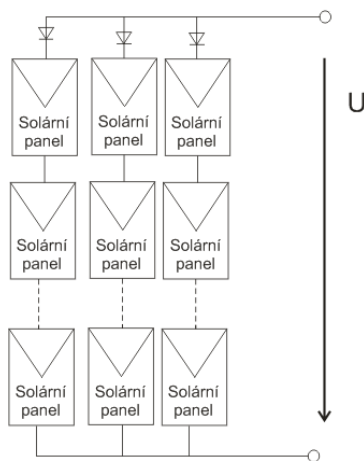
Paralelní řazení se provádí pro zvýšení maximálního odebíraného proudu. Při řazení je nutné zabránit zpětnému toku proudu mezi panely, k čemuž se používá sériová dioda u každého panelu. Tyto diody bývají v konvenčních solárních panelech integrovány.



Obr.:4 Paralelní řazení panelů

Kombinované řazení panelů

Jedná se o kombinaci sériového a paralelního řazení. Využívá se např. pro maximální využití určené plochy, dle použitého měniče a struktury systému se určí velikost jednoho řetězce, dále se tento řetězec opakuje na dané ploše kvůli zvýšení dodaného výkonu.



Obr.:5 Kombinované řazení panelů

Solární regulátor

Je "mozkem" celého solárního systému, jedná se o DC/DC měnič zajišťující bezpečné nabíjení solárního akumulátoru, plní funkci ochrany solárního panelu před přetížením, solárního akumulátoru před příliš velkým vybitím, přebitím, nabíjením vysokým proudem. Nejčastěji pracují ve formě snižujícího měniče, kdy napětí solárních panelů je vyšší než napětí akumulátoru, v takovémto případě dosahují účinnosti až 98%. Dnes jsou převážně řízeny pomocí jednočipových nebo malých signálových procesorů a využívají PWM regulace. Regulátory dle velikosti a výkonu umožňují nabíjet různé typy akumulátorů, umožňují progresivní nabíjení dle teploty a osvětlení panelu, komunikaci a ovládání pomocí komunikační linky (USB, Ethernet).

Základní rozdělení solárních regulátorů je dle poměru vstupního / výstupního napětí na zvyšující a snižující měniče. Další dělení je možné podle způsobu řízení a vybavenosti různými ochranami solárních panelů a solárních akumulátorů.

Usměrňovače

- Pouze usměrňují napětí ze solárního panelu, zabráňují zpětnému toku proudu do panelu
- Nejlevnější, nejjednodušší, nejmenší
- Využíváno pouze pro malé výkony mobilních zařízení
- Jednoduché nenáročné aplikace - solární lampičky, budíky, nabíječky v kalkulačkách

Snižující měnič

Jednoduché:

- Pouze stabilizace a omezení výstupního napětí
- Vyžadují větší osvětlení solárního panelu
- Jen částečné využití energetického potenciálu panelu
- Často využívají jen analogových komparátorů
- Levné, jednoduché, pro nenáročné aplikace

Řízené:

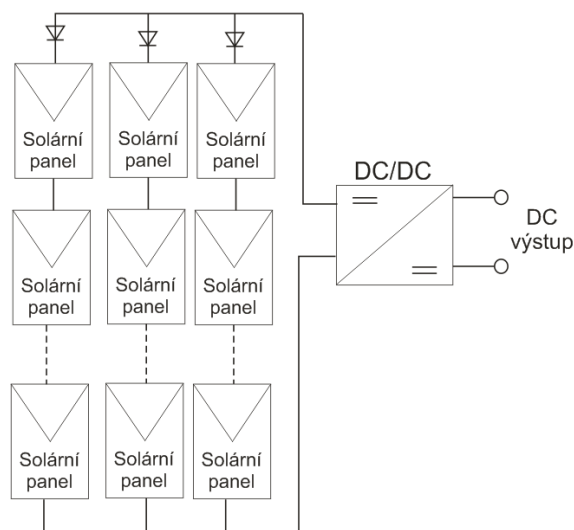
- Lepší ochrana akumulátorů a solárních panelů
- Možnost omezení nabíjecích proudů a udržovacího nabíjení (Standby)
- Vyžadují větší osvětlení solárního panelu
- Jen částečné využití energetického potenciálu panelu
- Řízené převážně jednočipovými procesory

Zvyšující měnič

- Lepší efektivita - využívá lépe energetický potenciál panelu
- Umožňuje "trackové" řízení - Viz. Kapitola 3
- Náročné na kvalitu měniče a řídicího systému

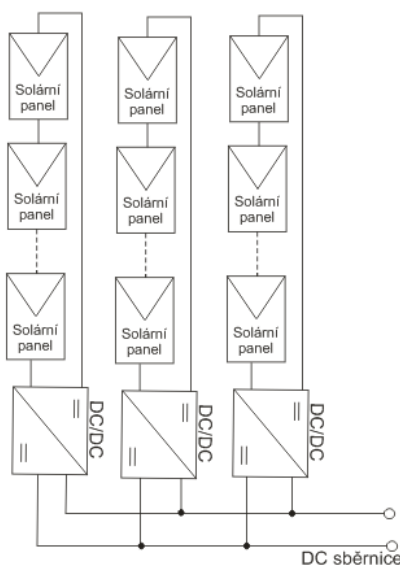
Řazení měničů

Řazení řetězců do jednoho pole se využívalo především v dřívějších dobách, kdy kvalita a především cena DC/DC měničů nedovolovala využití jejich většího počtu. Tento systém je málo efektivní především při různém osvětlení panelů.



Obr.:6 Spojení řetězců do jednoho pole

Dnes nejrozšířenějším způsobem je vytváření oddělených řetězců s vlastním DC/DC měničem. Tento způsob dovozuje lepší využití větších ploch s různou osvětleností.



Obr.:7 Jednotlivé řetězce s vlastními DC/DC měniči

Solární akumulátor

Je klíčovým prvkem celého systému, protože zajišťuje vyrovnaní energetické hladiny a dovoluje provoz zátěže i v době, kdy nejsou panely dostatečně osvětleny. Z pohledu provozu baterie se jedná o nepřetržitý cyklický provoz s nerovnoměrnou zátěží, což je pro baterii velmi náročné, výběr vhodné baterie je tedy klíčový s ohledem na dlouhou požadovanou dobu životnosti a minimální budoucí investice.

Druhy používaných akumulátorů:

Startovací baterie

Baterie určeny primárně pro startování vozidel. Jsou navrženy pro velmi vysoké proudy v krátkém čase. Pro solární systémy jsou nevhodné, protože zde dochází k dlouhodobému vybíjení nízkými proudy. Využívají se především při aplikaci solárních systémů na mobilní vozidla (karavany) nebo pro nenáročné zřídka využívané stavby (chata, srub), kde se bere ohled zejména na pořizovací cenu.

Trakční baterie

Využívají se především pro pohon malých elektrických zařízení (vozíků, lodí). Tyto akumulátory na rozdíl od startovacích jsou určeny pro cyklický provoz, jejich nevýhoda spočívá v nutnosti plného nabití před použitím. Pokud k plnému nabití nedojde, snižuje se kapacita a životnost akumulátoru. Hlavní výhodou je nízká pořizovací cena.

Staniční (záložní) baterie

Tyto baterie jsou určeny pro záložní zdroje a systémy např. v nemocnicích, elektrárnách, divadlech, kinech apod. Akumulátory se udržují neustále nabitě např. neustálým malým proudem a v případě výpadku elektrické energie jsou schopny okamžitě dodávat potřebný proud pro danou technologii. Pro solární systémy nejsou vhodné zejména z ohledu na cyklický provoz.

Solární baterie

Jsou určeny pro trvalý cyklický provoz v solárních systémech, vzhledově ani parametry se neliší od klasických baterií, rozdílné je složení aktivních elektrod uvnitř. Elektrody jsou uzpůsobeny, aby byly schopny dodávat dostatečný výkon i při částečně vybitém stavu.

Rozdělení podle provedení:

Uzavřené větrané:

Akumulátor má tekutý elektrolyt, který je potřeba jednou za čas dolít. Mají vyšší životnost oproti uzavřeným (VRLA) akumulátorům.

Ventilem řízené (VRLA):

Akumulátor má elektrolyt vázaný, je hermeticky uzavřený a nepotřebuje dodatečnou údržbu, jsou však určeny pro méně náročný provoz.

Ventilem řízené akumulátory se dále dělí:

AGM: Elektrolyt je vázán ve skelném rounu

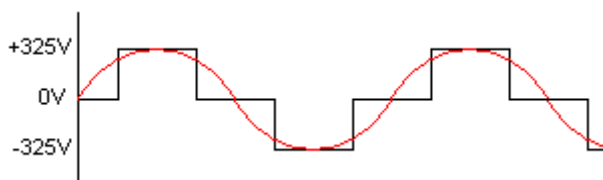
Gelové: Elektrolyt je ve formě gelu

Lithium-železo-fosfátové baterie (LiFePo):

Dnes velmi perspektivní baterie, které odbourávají neduhy olověných baterií (sulfatace elektrod, nutnost plného nabití). Nominální napětí jednoho článku je 3,2V, sériovým řazením čtyř článků dosáhneme baterie 12,8V. Umožňuje provoz při nižších teplotách bez ztráty parametrů, nabízí až 3x vyšší životnost oproti olověným akumulátorům při stejném zatížení. Baterie mají nízký vnitřní odpor a velkou hustotu energie (až 70% úspora hmotnosti a prostoru při stejné kapacitě). Vysoká účinnost až 92%, nižší citlivost na přebíjení. Tyto baterie však vyžadují nabíjení pomocí balancéru - jednotlivé články se nabíjejí odděleně a chrání je proti poškození. Balancér je součástí baterie a připojuje se k solárnímu regulátoru, který zajišťuje optimální nabíjení. Tyto moderní baterie mají vyšší pořizovací cenu, avšak s ohledem na kladné vlastnosti a mnohem delší životnost jsou správnou volbou pro všechny náročnější aplikace.

Měnič pro spotřební zařízení

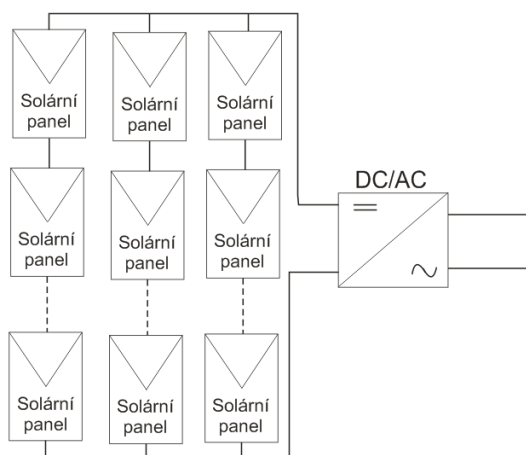
Jedná se o měnič DC/AC, nejčastěji pracující z 12/24V DC na 230V/50Hz AC. Je využívám pro zařízení neumožňující připojení na nízkonapěťový rozvod, např. lednička, televizor, elektrické ruční nářadí. Dle kvality a použité konstrukce je výstupní napětí ve formě modifikované sinusovky, nebo čisté sinusovky. Tento parametr je velmi důležitý především u moderní citlivé elektroniky (LCD televize, LED osvětlení, měřicí přístroje) a pro napájení elektromotorů (kompresory, oběhová čerpadla).



Obr.:8 Rozdíl mezi čistou a modifikovanou sinusovkou [14]

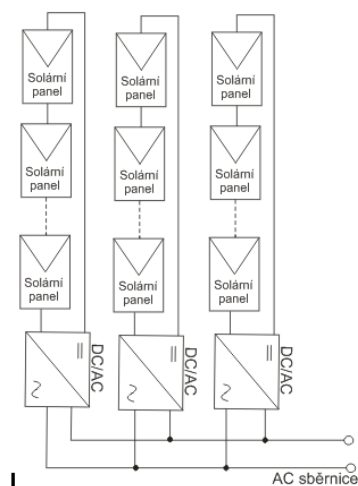
Řazení měničů

Připojení měničů je určeno především využitím systému. U jednoduchých, levných, často mobilních zařízení se využívá přímé přeměny energie dodané solárními panely na střídavé napětí.



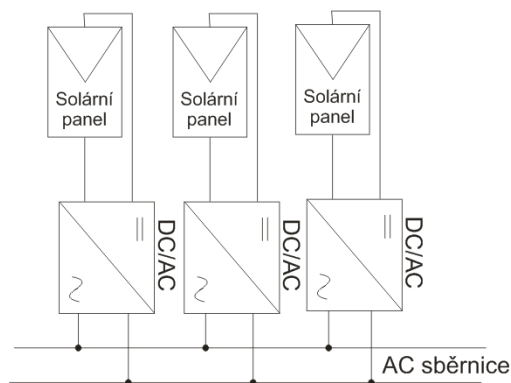
Obr.:9 Přímé připojení DC/AC měniče k solárnímu poli

Obdobně jako u řazení DC/DC měničů se pro lepší využití velké plochy používá většího počtu měničů, každý pro jednotlivý řetězec.



Obr.:10 Připojení jednotlivých měničů k řetězcům

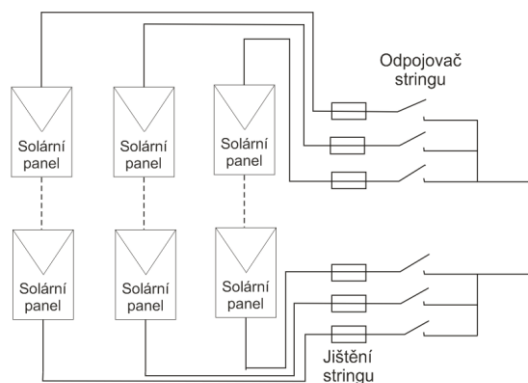
Díky zmenšování spínacích součástek a zvyšování kvality měničů se dnes pro zvláštní případy používá i integrace měničů přímo do solárních panelů. Tyto panely pak mají AC výstup a jejich spojování je realizováno paralelní sběrnicí.



Obr.:11 Integrace měničů pro jednotlivé solární panely

Jištění fotovoltaického systému

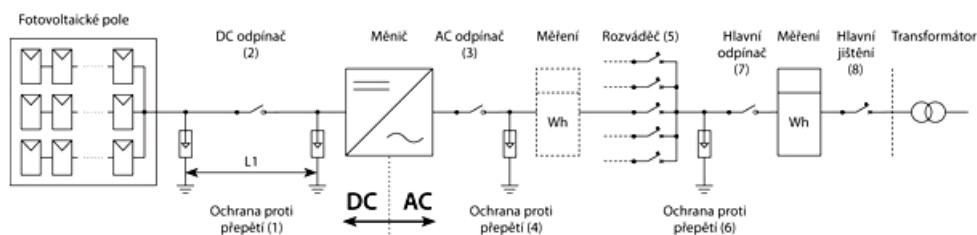
Jištění jednotlivých řetězců je nutné na obou výstupních svorkách, především kvůli ochraně před zpětnými proudy, nadproudy a ochraně kabelů při poruše. Toto jištění je především v levnějších a jednodušších systémech opomíjeno z důvodu nízkého zkratového proudu, který jen o 10-20% převyšuje jmenovitý proud.



Obr.:12 Způsob jištění řetězců [7]

Připojení fotovoltaického systému k rozvodné síti

Z Obr.:13 je patrný způsob připojení fotovoltaického systému k rozvodné síti. Protože se u stabilních řešení jedná o zařízení umístěné především ve venkovním prostředí, je nutné dbát na správné přepětěvé jištění všech uzlů doplněné dle umístění i hromosvody a zemněním. Dále je celý systém doplněn DC a AC odpínači z důvodu údržby a měřiči vyrobené a dodané elektrické energie.



Obr.:13 Způsob připojení fotovoltaického systému k rozvodné síti [7]

Návrh řídicí jednotky DC/DC měniče

Práce je zaměřena na fotovoltaický systém obsahující tři jednotlivé 12V panely na vstupu a olovený bezúdržbový akumulátor na výstupu s napětím 96V. Pro tuto soustavu byl určen jednoduchý zvyšující měnič pracující pro vstupní napětí 0-50V s výstupním napětím 50-100V.

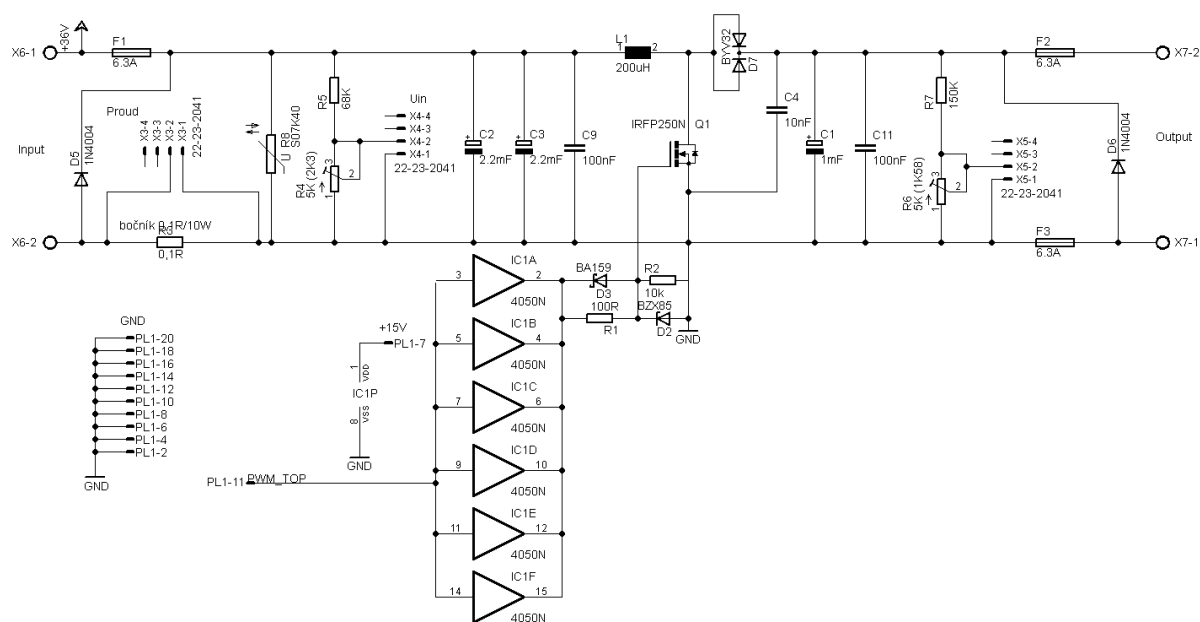
Zvyšující DC/DC měnič

Jako měnič byl použit jednoduchý zvyšující měnič z diplomové práce Bc. Tomáše Chalupky [5]. Měnič je přímo určený pro použití v solárním systému. Měnič je vybaven vstupní i výstupní proudovou ochranou tvořenou tavnými pojistkami, vstup řídicího signálu je ošetřen pomocí budiče CMOS 4050.

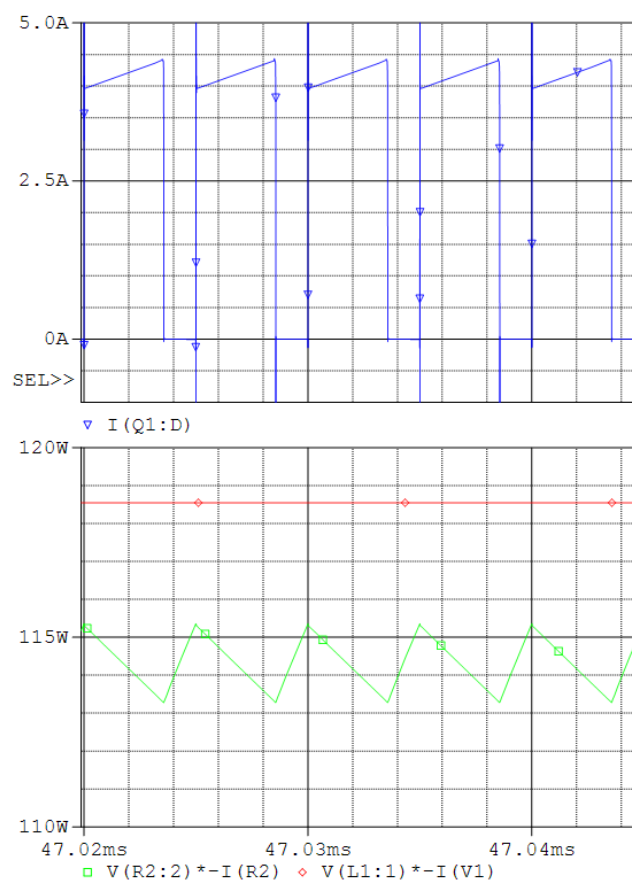
Výstupní napětí je určeno vztahem:

$$U_{OUT} = U_{IN} \cdot \frac{1}{1-\delta} \quad (1)$$

Střída - δ (činitele plnění)

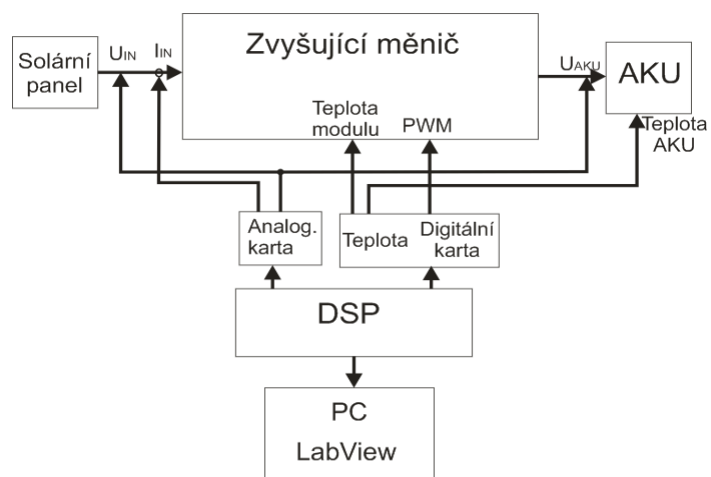


Obr.: 14 Schéma zvyšujícího měniče



Obr.:15 Simulace měniče

(modrá - proud výkonovým tranzistorem, červená - vstupní příkon, zelená - výstupní zvlněný výkon)



Obr.:16 Blokové schéma připojení řídicího systému k měniči

Vývojový kit TMS320F28335

Pro řízení měniče byl zvolen vývojový kit se signálovým procesorem Texas Instruments TMS320F28335. Jedná se o univerzální modul obsahující komunikační rozhraní USB, SPI, SCI, RJ45, programovací rozhraní JTAG, PWM výstupy, analogové vstupy a výstupy, vstupy pro teplotní čidla a relé výstup.

Základní parametry TMS320F28335:

- Frekvence procesoru až 150MHz
- Napájení 1,8/1,9V pro procesor, 3,3V logika
- 32 bitová sběrnice s plovoucí desetinnou čárkou
- 16 / 32 bit vnější rozhraní
- 18 PWM výstupů, nebo 6 PWM výstupů s vysokým rozlišením
- 3 32bit časovače
- 12bit AD převodník se 16 kanály (2x8 kanálů MX), vstupní napětí 0-3V
- Pracovní teplota -40°C až 80°C

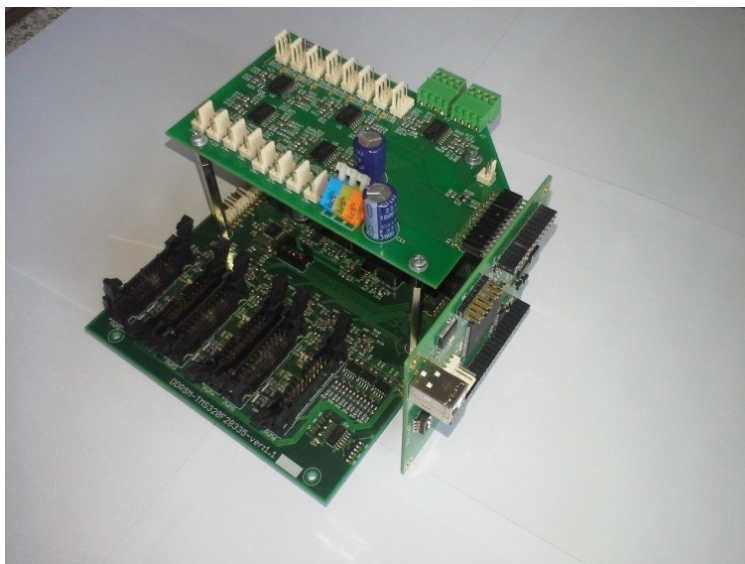
Napájení modulu je rozděleno na digitální část s napájením 24V a analogovou část s napájením $\pm 15V$. Odběr celého kitu se pohybuje okolo 5W bez připojených čidel.

Sestava kitu se skládá ze tří vzájemně propojených desek: Procesorová deska, Digitální deska a Analogová deska.

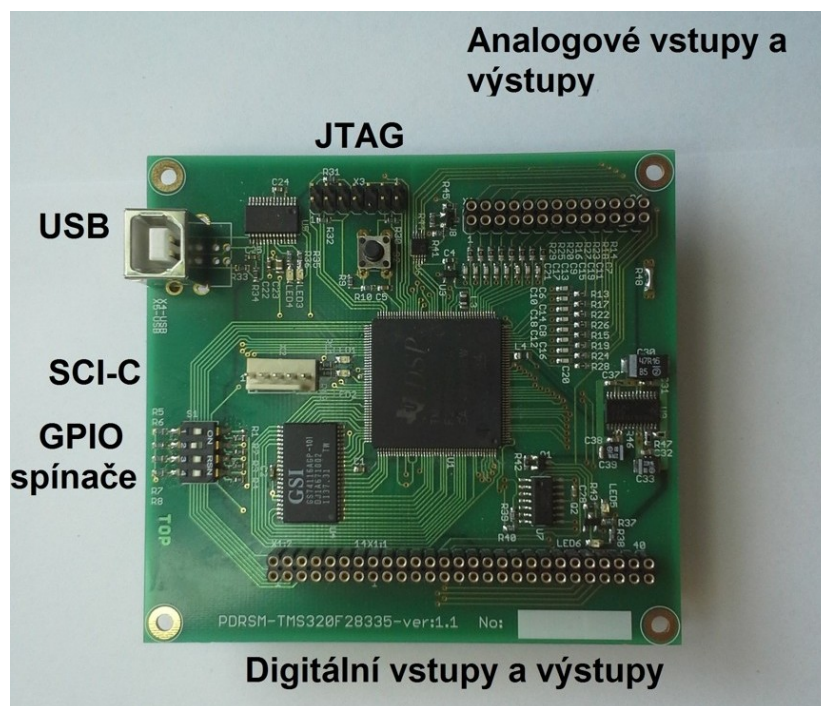
Procesorová deska: Obsahuje signálový procesor, USB, SCI a JTAG rozhraní

Digitální deska: Obsahuje 6 dvojic PWM výstupů včetně chybových vstupů z budiče, rozhraní RJ45, SPI rozhraní, relé výstup, 6 vstupů pro teplotní čidla, 2 vstupy pro čidla průtoku, zdrojovou část pro napájení digitální desky a procesorové desky

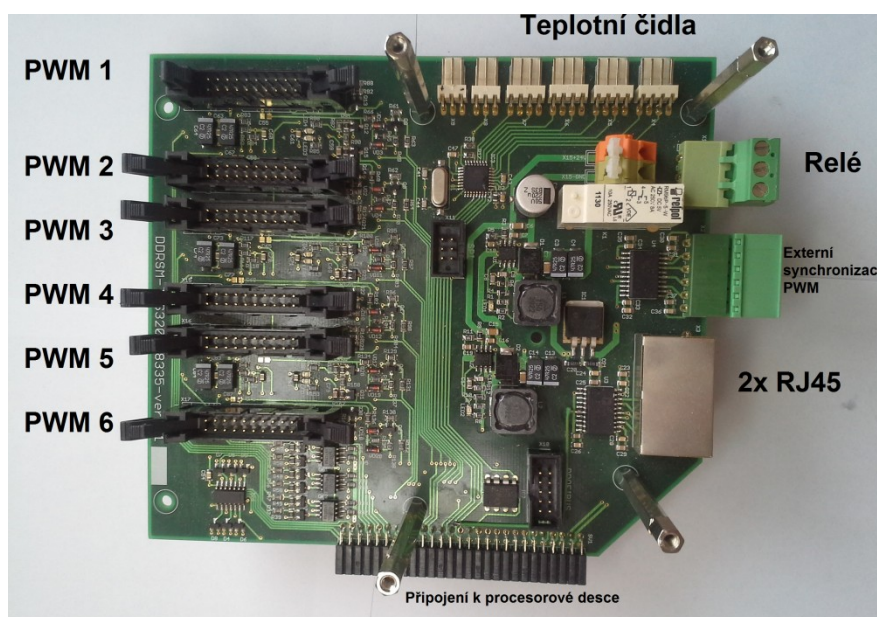
Analogová deska: Obsahuje 16 analogových vstupů, 4 analogové výstupy, zdrojovou část pro napájení analogových obvodů



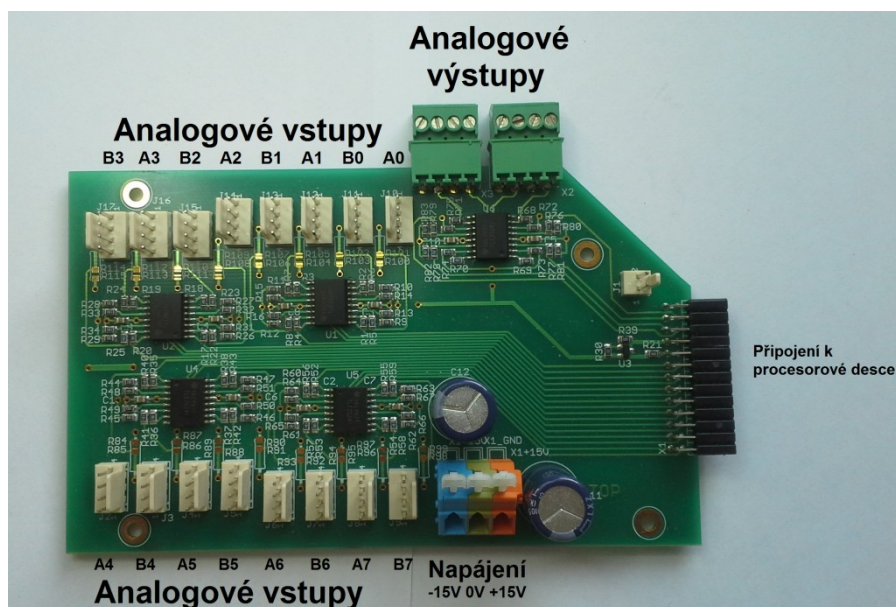
Obr.:17 Vývojový kit



Obr.:18 Procesorová deska



Obr.:19 Digitální deska



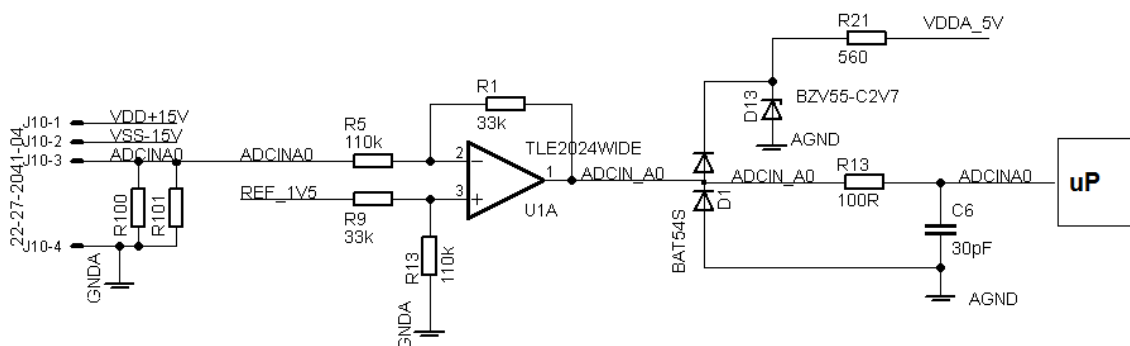
Obr.:20 Analogová deska



Obr.:21 Programátor C2000

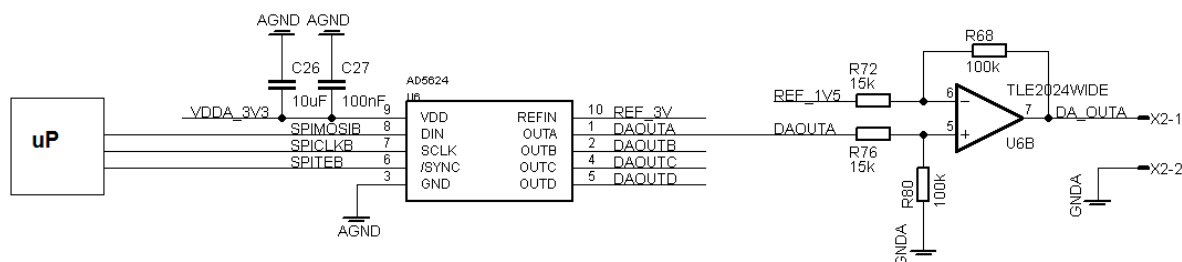
Popis zapojení jednotlivých částí:

Analogové vstupní napětí může dosahovat velikosti $U_{IN} = -5V \div 5V$, analogový vstup procesoru je ošetřen proti vstupnímu přepětí.



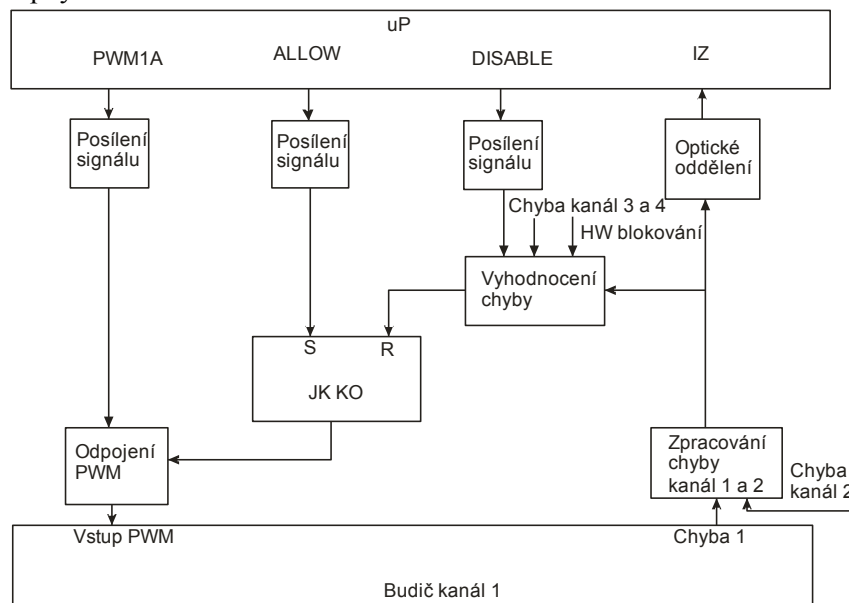
Obr.:22 Zapojení analogových vstupů (Vstup ADCINA0)

Analogový výstup není ošetřen proti zkratu a může být zatěžován maximálně proudem $I_{DAOUTMAX} = \pm 40mA$, výstupní napětí $U_{DAOUT} = -10V \div 10V$.



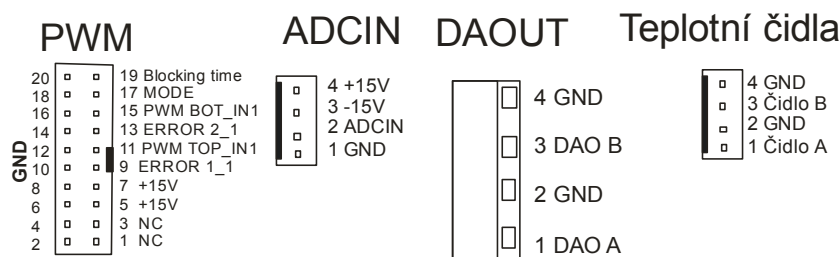
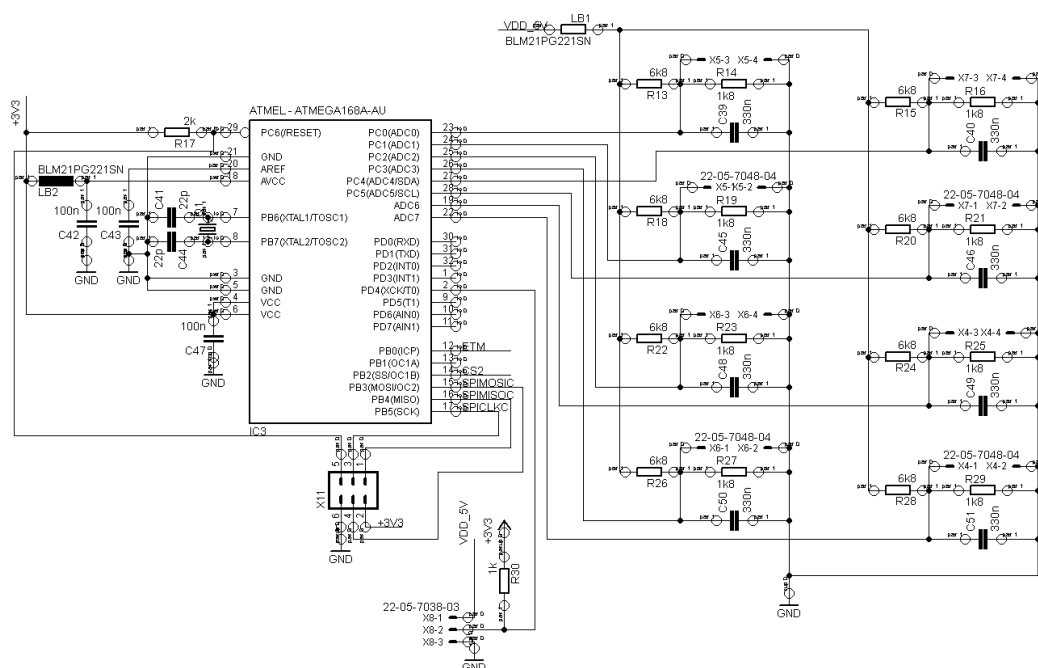
Obr.:23 Zapojení analogových výstupů (Vstup DAOUTA)

PWM výstupy jsou uspořádány do šesti párů určených pro řízení "H" můstků pro tři jednotlivé fáze. Každá dvojice je proto označena BOT a TOP, umožňuje programové nastavení mrtvé doby komplementární dvojice tranzistorů. Každý PWM výstup má přiřazen jeden chybový vstup ERROR, tyto chyby jsou přenášeny zpětně do mikroprocesoru jako signál IZ. Z Obr.:24 vyplývá, že vzniklá chyba kdekoliv v jedné fázi zablokuje PWM signál pro celou fázi. Po odstranění zdroje chyby je následně potřebné programové odblokování pomocí posloupnosti signálů ALLOW a DISABLE, jež odblokuje JK klopný obvod.



Obr.:24 Blokové schéma zapojení PWM výstupu a chybových vstupů

Blok vyhodnocení teploty je řízen mikroprocesorem ATMEGA 168, ten obsahuje program pro převod teplotních charakteristik měřících čidel na teplotu, plní tedy pouze funkci automatického převodníku se sériovou komunikací po lince SPI.

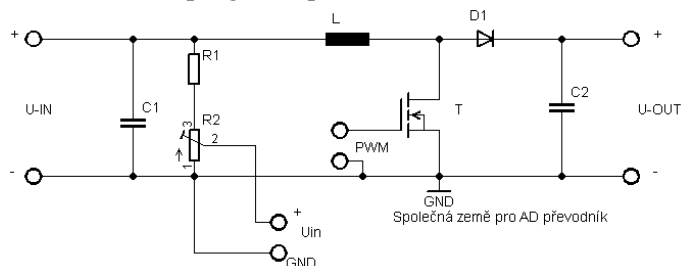


Výpočet hodnot děliče:

Vstupní napětí $U_I=0-50V$, zvolen $R_I=100k\Omega$, $U_D=5V$

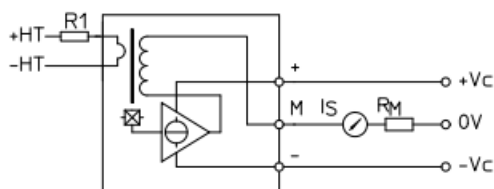
$$U_D = U_1 \frac{R_2}{R_1+R_2} \rightarrow R_2 = \frac{U_D R_1}{U_1 - U_D} \cong 12k \quad (2)$$

Pro odpor R_2 byl zvolen více otáčkový trimr, dovolující přesnější nastavení dělicího poměru. Případná další odchylka se odstraní v programu procesoru.



Obr.:27 Zjednodušené schéma zvyšujícího měniče s připojeným napěťovým děličem

Pokud bude vyžadováno galvanické oddělení měřeného napětí, je možné využít integrovaných čidel např. LM-25P firmy LEM. Tyto čidla vyžadují souměrné napájecí napětí $V_C = \pm 12V$ a jejich výstupem je velikost a směr proudu, měřeného na odporu R_M , odpovídající velikosti napětí na vstupu viz Obr.:28



Obr.:28 Čidlo LM-25P, vnitřní struktura

Měření proudu

Pro měření proudu byl s ohledem na nízký využívaný proud zvolen proudový bočník, realizovaný pomocí výkonového rezistoru $0,1\Omega/10W$. Toto řešení bylo zvoleno především kvůli jednoduchosti vývojového kitu, pro hotový modul by bylo využito proudové čidlo s hallovým senzorem.

Proudový bočník:

Měřicí bočník je zapojen v záporné větvi měniče, s ohledem na realizovaný měřicí AD převodník, kvůli vytvoření společného zemního bodu. Dle zadání práce by měl měnič pracovat se výstupním napětím do $100V$ a výkonem do $100W$. Při využití bočníku s hodnotou $R_B=0,1\Omega/10W$ bude maximální napětí na bočníku

$$U_{RB} = R_B * I_{MAX} = 0,1 * 10 = 1V \quad (3)$$

výkonová ztráta na tomto bočníku

$$P_{MAX} = U_{RB} * I_{MAX} = 1 * 10 = 10W \quad (4)$$

Měření teploty

Měření neelektrických veličin jako teplota nevyžaduje galvanické oddělení přívodních vodičů, protože to je realizováno už principem měření. Měření teploty je primárně využito pro teplotní kompenzaci napětí při nabíjení akumulátoru a dále pro ochranu spínacích součástí měniče proti přehřátí.

Vývojový kit TMS320F28335 je vybaven převodníkem umožňujícím měření pomocí až 6 odporovými teplotními čidly, bez nutnosti dalších periferních obvodů. Toto měření bylo původně předurčeno pro IGBT moduly POWEREX CM200DX-24S s integrovanými teplotními NTC čidly. Díky převodníku je v programu přímým výstupem hodnota teploty. NTC čidla jsou typu s vnitřním odporem $R_{25}=5k\Omega$ při $25^{\circ}C$. Teplotní čidla se připojují v párech pomocí konektorů X4-X7 dle Obr.:25 a Obr.:26.

Řízení nabíjení akumulátoru

Akumulátory využívané především v mobilních v solárních aplikacích jsou tzv. bezúdržbové (viz. Strana 15). Tyto akumulátory mají elektrolyt vázaný v gelu nebo skelné tkanině, což znamená, že nevyžadují údržbu ze strany dolévání elektrolytu. Každý, i sebekvalitnější, akumulátor však podléhá samovybití, což vyžaduje občasnou kontrolu a případné dobití. Obvyklá doba samovybití až na hranici zničení je kolem 9 měsíců. Faktory ovlivňující tuto dobu jsou kvalita a stáří akumulátoru, jeho stav, skladovací teplota a kapacita akumulátoru. U nově zakoupeného akumulátoru je tato doba mnohem vyšší, protože je od výroby připraven na delší skladování před uvedením do provozu.

Charakteristika U

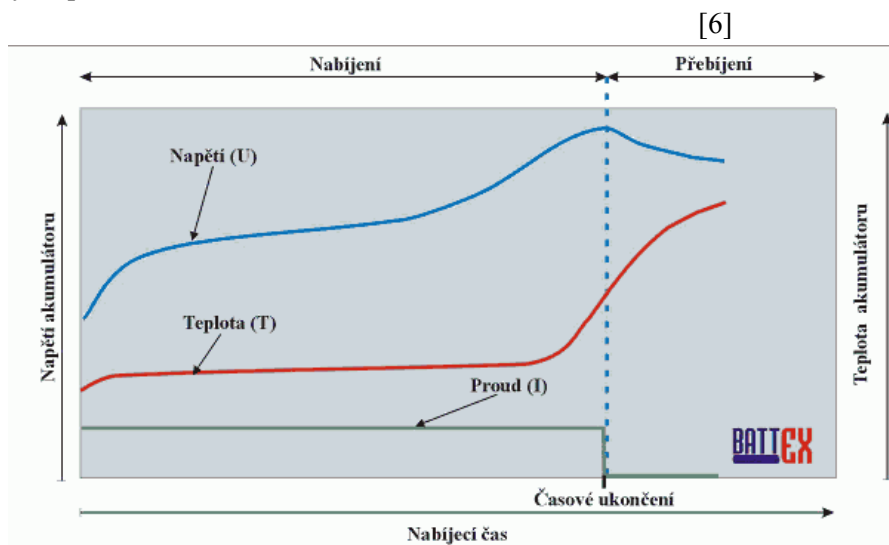
Nabíjení konstantním napětím, jehož hodnota se nastavuje na „plynovací napětí“ akumulátoru, tedy na cca 14,4 až 14,8V. Po připojení nabíječky tohoto typu je zahájeno nabíjení značně velkým proudem (jeho velikost bude závislá na hloubce vybití akumulátoru), který se konstrukčně zpravidla omezuje na max. 0,5 až asi 1 x násobek jmenovité kapacity akumulátoru.

Jak se napětí akumulátoru při nabíjení zvyšuje, klesá dobíjecí proud. Konečný nabíjecí proud dosahuje asi 0,002 násobku jmenovité kapacity akumulátoru. Vybitý akumulátor se takto nabije za 10 až 15 hodin, však už za 1 až 2 hodiny je akumulátor nabit na 80% konečné kapacity. Doporučenou nutností je už v počáteční fázi nabíjení hlídat teplotu elektrolytu, neboť díky velkým protékajícím proudům dochází k značnému vývinu tepla. Doporučuje se nepřekročit $40^{\circ}C$. Výhodou nabíjení dle této charakteristiky je rychlost nabíjení a určitá volnost ke konci nabíjení. Při dostatečném poklesu nabíjecího proudu je nutné nabíjení ukončit. [6]

Charakteristika I

Dobíjení konstantním proudem, jehož hodnota se nastavuje zpravidla na 0,08 až 0,1 násobek jmenovité kapacity akumulátoru. Vybitý akumulátor takto plně nabijeme za asi 10 až 14 hodin, přičemž zde neexistuje „efekt“ rychlého nabití jako u nabíjení konstantním napětím. Výhodou je snadný výpočet dodaného náboje. Stačí znát dobíjecí proud a čas, po který nabíjíme. Nevýhodou je fakt, že konečný dobíjecí proud je stejně veliký, jako iniciační. Hrozí tedy značné přebíjení a rychlé

„vyvaření“ akumulátoru, pokud není nabíječka vybavena automatikou hlídající stav plného nabití, či alespoň časovým spínačem.



Obr.:31 Charakteristika proudového nabíjení

Charakteristika W

Dobíjení se zvyšujícím se napětím a klesajícím proudem. V průběhu nabíjení se zvyšuje hodnota napětí nabíječky. Však díky zvyšování svorkového napětí akumulátoru klesá nabíjecí proud, ne však tak strmě, jako u nabíjení konstantním napětím. Nabíječka s touto charakteristikou proto nabíjejí rychle, neboť pracuje s poměrně velkými proudy v průběhu celého nabíjecího procesu. [6]

Charakteristika UI

Jedná se o kombinaci výše popsaných charakteristik. Nabíjení začíná konstantním proudem, přičemž se hlídá dosažení plynovacího napětí, kdy se přechází v nabíjení konstantním napětím a dále v udržovací nabíjení sníženým napětím (Standby).

VRLA akumulátor s Gelovým elektrolytem

Nabíjecí proces je zde více kritický, neboť se jedná o hermeticky zapouzdřený akumulátor s vnitřním rekombinačním cyklem, takže jej nelze nabíjet jako klasické akumulátory se zaplavenými elektrodami. Základní podmínkou je, že akumulátor nesmíme nabíjet napětím převyšujícím hodnotu plynovacího napětí a dále, po nabití akumulátoru nabíjení ihned ukončíme, tedy pokud nepřecházíme na nabíjení udržovací (Standby), sníženým napětím. Přebíjení gelových akumulátorů zkracuje jejich životnost, v některých případech může skončit i výbuchem.

U gelových akumulátorů je tedy absolutním pravidlem, že napětí na svorkách nabíječky nesmí překročit cca 14,4V (po krátkou dobu, z počátku nabíjení, lze připustit i vyšší napětí k oživení vybité baterie), vše se řídí především specifikacemi jednotlivých výrobců. V opačném případě dochází ke zkrácení životnosti akumulátoru.

Abychom zajistili bezpečnost nabíjení, využívá se při nabíjení gelových akumulátorů tzv. „IU“ nabíjecí charakteristika. To znamená, že nabíjení zahajujeme konstantním proudem

(I) o velikosti 0,1 jmenovité kapacity, po dosažení 14,1 až 14,4V přepínáme na nabíjení konstantním napětím (U) o velikosti 14,1 až 14,4V, kdy teče do akumulátoru již malý proud. Následovat může i fáze udržovací, kdy se dobíjí konstantním napětím sníženým na cca 13,6 - 13,8V. [6]

VRLA akumulátor s vázaným elektrolytem AGM

Proces nabíjení je prakticky shodný s gelovými akumulátory. Opět se jedná o hermeticky zapouzdřené akumulátory s vnitřním rekombinačním cyklem se všemi důsledky z toho plynoucími, včetně rizika destrukce při přebíjení. U AGM akumulátorů je absolutním pravidlem, že dobíjecí napětí nesmí překročit 14,6V (tedy zde se připouští o něco vyšší napětí než u akumulátorů gelových, někteří výrobci připouštějí až 14,8V).

Abychom zajistili bezpečnost nabíjení, využívá se rovněž při nabíjení AGM akumulátorů tzv. „IU“ nabíjecí charakteristika. To znamená, že nabíjení zahajujeme konstantním proudem (I) o velikosti 0,1 jmenovité kapacity, po dosažení 14,4 až 14,6V přepínáme na nabíjení konstantním napětím (U) o velikosti 14,4 až 14,6V kdy teče do akumulátoru již malý proud. Následovat může i fáze udržovací, kdy se dobíjí konstantním napětím sníženým na cca 13,6 - 13,8V. [6]

Přibližný stav nabití pro hermetické AGM/gelové 12V akumulátory:

(akumulátor bez připojené zátěže, nebo nabíjení minimálně 12h)

12,8V = 100%

12,5V až 12,6V = 75%

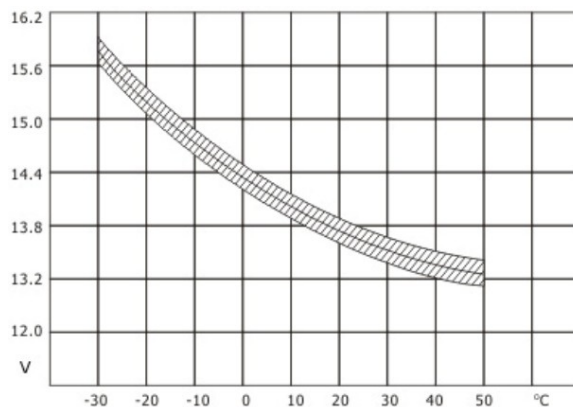
12,2V až 12,3V = 50%

12,0V až 12,1V = 25%

11,8V = vybitý

Teplotní kompenzace

Dosahovaná napětí na svorkách akumulátoru jsou deklarována obvykle pro 25°C. Pro zvýšení životnosti akumulátoru a zajištění bezporuchového provozu je vhodné přepočítat napětí určitým kompenzačním koeficientem, přičemž čím nižší teplota, tím vyšší napětí a naopak. Přibližně lze uvažovat, že na každých 10°C teplotní odchylky (teplota akumulátoru – elektrolytu), upravíme dobíjecí napětí přibližně o cca 0,3V (tedy asi 0,03V/1 °C - platí v případě 12V akumulátoru - 6 článků). Pokud pracujeme za nižší teploty, upravíme napětí směrem nahoru, pokud za vyšší, upravíme napětí směrem dolů. [6]



Obr.:32 Teplotní kompenzace nabíjecího a udržovacího napětí [6]

Vytvoření programu pro řídicí jednotku

Pro řídicí jednotku bylo udáno několik základních požadavků s ohledem na její budoucí určení. Cílem bylo maximálně využít jak možností vývojového kitu, tak parametrů měniče, pro který je systém určen.

- Pracovní frekvence PWM při automatickém řízení 200kHz
- Vstupní napětí 0-50V (pro použití 3 sériově řazených 12V panelů)
- Výstupní napětí do 150V (pro připojení 96V olověného bezúdržbového akumulátoru)
- Trackové řízení
- Teplotní ochrana měniče a akumulátoru
- Možnost manuálního řízení
- Komunikační rozhraní LabView

K vývojovému kitu [TMS320F28335](#) je k dispozici se základní kostra programu v C++ obsahující především inicializaci periférií a nastavení komunikačního rozhraní s LabView. Dále je součástí ukázkové rozhraní v LabView ukazující některé možnosti zobrazení a zadávání dat.

Pro samotné vytváření programu je využito vývojové prostředí "Code Composer Studio v4" (CCSv4). Program umožňuje automatické nahrání programu do procesoru ihned po kompilaci, spolupracuje s programátorem C2000, jež je součástí vývojového kitu. Programátor se připojuje k PC přes rozhraní USB. Programátor nevyžaduje instalaci ovladačů. Programování probíhá automaticky po nastavení programátoru v "debuggeru" a spuštění kompilovaného programu.

Při vytvoření nového programu je nutné zadat cestu pro zdrojové adresáře knihoven. Dodatečné přidávání jednotlivých knihoven se provádí pomocí kontextového menu projektu. Dále je ukázáno několik důležitých bodů programu s vysvětlením jejich funkcí a principu, kompletní program včetně komentáře je k dispozici v textové a datové příloze.

Vložení knihoven:

Knihovny definují základní vlastnosti procesoru, jeho periferie, využívané datové typy, matematické operace.

```
#include "DSP28x_Project.h" //knihovna procesoru DSP28xx
#include "ENET.h"           //Definuje všechny vnější přerušení (SPI, ext. RAM, DAC...)
#include "regmacro.h"       //Definuje vektorové operace.
#include "math.h"           //Definuje základní matematické operace.
```

Definice proměnných: (Ukázka několika základních datových typů)

Při určení datového typu proměnné se bere v potaz hodnota, kterou bude proměnná v programu nabývat, např. pro proměnnou v čekací smyčce "FOR" bude postačovat datový typ "Uint

16/32 (Unsigned Integer)" tedy bezznaménková forma nabývající pouze kladných hodnot v rozsahu (Uint 16) 0-65535. Pro výpočty reálných hodnot např. z AD převodníku se využije datová proměnná typu "Float 32", což je 32 bitové reálné číslo s plovoucí desetinnou čárkou.

Dalším krokem je určení počáteční hodnoty dané proměnné, ta se odvíjí především od funkce proměnné v programu a měla by vyjadřovat ideální ustálený stav. Hodnota takto určených proměnných může být v programu dále měněna.

```
Uint16  MODE = 0;           //MODE 0=manuál, 1 = auto
float32  Strida_PWM = 50;    //Střída PWM 50%
Uint32   timeout_scia = 0;
```

Pomocí funkce "define" můžeme určit i konstanty, jejíž hodnota bude pevně nastavena a v programu už nebude moci být změněna, např. konstanta π

```
#define PI      3.141592654      //definice proměnné PI
```

Deklarace funkcí v programu:

Jedná se především o funkce obsluhy jednotlivých přerušení.

```
interrupt void adc_isr(void);    //Přerušení od AD převodníku
```

Hlavní smyčka:

Inicializace systémových součástí (PLL, WatchDog, Peripheral Clocks, atd.)

```
InitSysCtrl();    //INITialize SYStem ConTRoL
```

Inicializace GPIO (General Purpose Input Output)

```
EALLOW;           //povolení zápisu do registrů

GpioDataRegs.GPASET.bit.GPIO30 = 1;
GpioDataRegs.GPBSET.bit.GPIO34 = 1;
GpioCtrlRegs.GPAMUX2.bit.GPIO30 = 0;
GpioCtrlRegs.GPADIR.bit.GPIO30 = 1;      //RELE1
GpioCtrlRegs.GPBMUX1.bit.GPIO34 = 0;
GpioCtrlRegs.GPBDIR.bit.GPIO34 = 1;      //RELE2
GpioCtrlRegs.GPBMUX1.bit.GPIO32 = 2;      //PWMSYNCCIN
GpioCtrlRegs.GPBQSEL1.bit.GPIO32 = 2;
GpioCtrlRegs.GPBDIR.bit.GPIO32 = 0;
GpioCtrlRegs.GPBMUX1.bit.GPIO33 = 2;      //PWMSYNCCOUT
GpioCtrlRegs.GPBDIR.bit.GPIO33 = 1;

EDIS;           //ukončení zápisu do registrů
```

Zakázání všech přerušení CPU

Bez tohoto příkazu bychom nebyli schopni zaručit správné zavedení programu, protože přichází přerušení během inicializace PIE, a periférií by způsobila pád celého programu.

```
DINT; //Disable INTerrupt
```


Inicializace PIE registru na výchozí stav

Všechny PIE přerušení zakázány a příznaky vymazány.
InitPieCtrl(); //INITialize PIE ConTRoL

Zakázání přerušení CPU a vynulování všech příznaků

IER = 0x0000;
IFR = 0x0000;

Inicializace všech PIE vektorů přerušení na ukazatele ISR(Interrupt Service Routines)

Toto pak usnadní případné debugování.
InitPieVectTable();

Přemapování vektoru přerušení

EALLOW; // povolení zápisu do chráněného registru
PieVectTable.ADCINT = &adc_isr;
EDIS; //konec zápisu do registrů

Inicializace periférií:

InitGpio(); //GPIO
InitSpiB(); //SPI B
InitSpiBGpio();
InitSpiC(); //SPI C
InitSpiCGpio();
InitScia(); //SCI A
InitScib(); //SCI C
InitAdc(); //ad převodník
ConfigAdc(); //AD převodník nastavení
InitExRAM(); //externí RAM

Inicializace PWM a nastavení počátečních hodnot pro všech 6 kanálů:

InitEPwm(PWM_MAX, PWM_HSD, PWM_DIV, PWM_FED, PWM_RED);
EPwm1Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * 0)+PWM_MAX2);
EPwm2Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * 0)+PWM_MAX2);
EPwm3Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * 0)+PWM_MAX2);
EPwm4Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * 0)+PWM_MAX2);
EPwm5Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * 0)+PWM_MAX2);
EPwm6Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * 0)+PWM_MAX2);

Povolení přerušení pro skupinu 1 - AD převodník a Timer 0:

IER |= M_INT1;

Povolení přerušení od ADC v PIE

PWM bude vyvoláno přerušením od AD převodníku (zajistí změnu hodnoty PWM až po načtení celé hodnoty z AD převodníku)

PieCtrlRegs.PIEIER1.bit.INTx6 = 1;

Povolení všech přerušení

```
EINT; // Povolení všech přerušení INTM
ERTM; // Povolení přerušení realtime debuggeru DBGMC
```

Inicializace AD převodníku

```
McbSpaRegs.DXR2.all=0x0028;
McbSpaRegs.DXR1.all=0x0001;
while(McbSpaRegs.SPCR2.bit.XRDY == 0){
    for(i = 0; i<256; i++)
        { pole_scia[i] = 0x00FF; }
    init_crc16_tab();
    neco = 0x12345678;
    GpioDataRegs.GPACLEAR.bit.GPIO30 = 1;
```

Nekonečná smyčka:

1. Zajišťuje komunikaci s LabView na PC prostřednictvím USB portu. Viz. Příloha: Program C++
2. Zajišťuje povolování a zakazování pulzů z PWM výstupů.
3. Zajišťuje čtení teploty (komunikace s ATmega 168). Viz. Příloha: Program C++

Ad.:2) Povolení a zákaz pulzu se provádí změnou chráněných registrů - je nutné programově uvolnit zápis do registru a po změně hodnoty registr opět uzavřít. Příklad pro povolení a zakázání pulzu kanálu 1. Toto je využito například při fungování teplotních ochran, kdy je jednoduše možné vypnout celý PWM výstup.

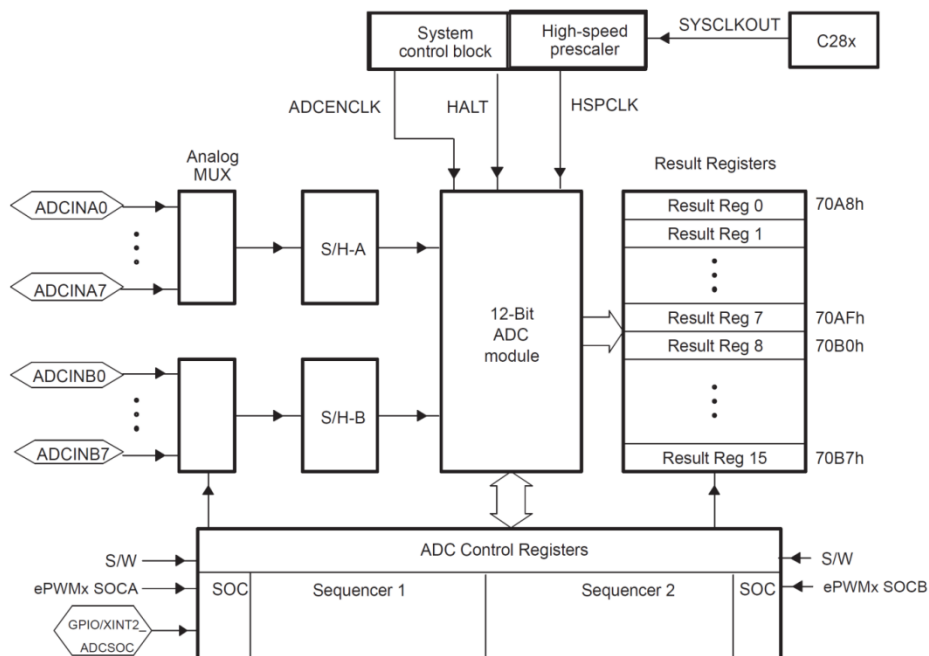
```
if((PWMon & 0x0001) > 0)
{
    PWMon -= 0x0001;
    EALLOW;
    EPwm1Regs.TZCLR.bit.OST = 1;
    EPwm1Regs.TZCLR.bit.CBC = 1;
    EDIS; }
if((PWMOff & 0x0001) > 0)
{
    PWMOff -= 0x0001;
    EALLOW;
    EPwm1Regs.TZFRC.bit.OST = 1;
    EDIS; }
```

Přerušení:

Toto přerušení je vyvoláno v polovině pulzu generovaného PWM a je určeno pro načtení hodnot z AD převodníku- tím se zajistí čtení hodnot v ustáleném stavu. Na Obr.: 33 je vidět rozdělení AD převodníku na dva 8 kanálové bloky pro obsluhu ePWM modulů, tyto bloky lze zařadit kaskádově a vytvořit 16 kanálový modul. Spuštění převodu je možno provést třemi způsob

- Externí spuštění
- Spuštění výstupem PWM generátoru
- Spuštění pomocí Softwaru

Převáděný kanál je vybírán pomocí multiplexoru, dále uložen v analogové paměti a dle potřeby postupně převeden v jediném AD převodníku. Tímto způsobem je možné měřit současně dva signály (jeden z kanálu A a druhý z kanálu B).



Obr.:33 Blokové schéma AD převodníku

Manuální nastavení hodnot PWM z LabView

Manuální režim LabView umožňuje ruční změnu střidy a frekvence PWM signálu. Střidu je možno měnit přímým přístupem do registru (PWM_data[x]). Frekvence PWM je určena registrem PWM_MAX při jehož změně je nutné manuálně zakázat a povolit PWM pulzy.

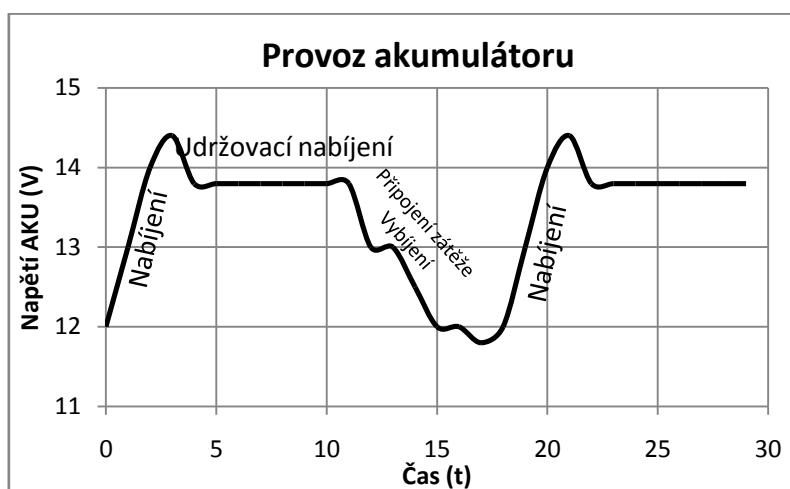
```

if (PWM_MAX != PWM_lab)    //Podmínka při změně hodnoty frekvence z LabView
{
    PWM_MAX=PWM_lab;        //Nastavení proměnné PWM_MAX na změněnou hodnotu
    PWM_MAX2=PWM_MAX/2;     //Výpočet PWM_MAX2
    PWM_PRC=PWM_MAX/100;    //Výpočet PWM_PRC
    InitEPwm(PWM_MAX, PWM_HSD, PWM_DIV, PWM_FED, PWM_RED);
    //Inicializace PWM
    EPwm6Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * 0)+PWM_MAX2);
    //Změna na nastavenou hodnotu
    EALLOW;                //Povolení zápisu do registru
    EPwm6Regs.TZFRM.bit.OST = 1; //zakázání pulzu
    EPwm6Regs.TZCLR.bit.OST = 1; //povolení pulzu
    EPwm6Regs.TZCLR.bit.CBC = 1; //povolení pulzu
    EDIS;                  //Zakázání zápisu do registru
}

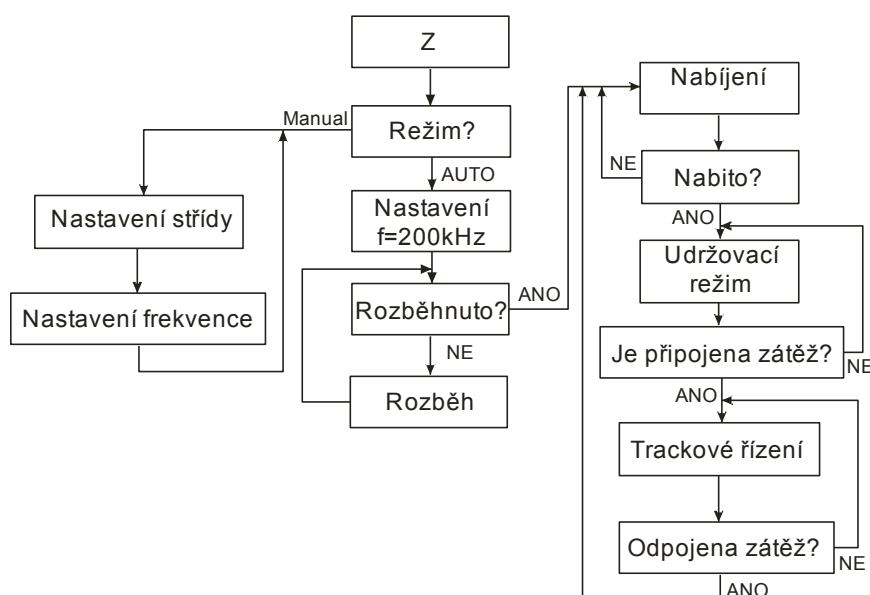
```

Regulační smyčka měniče

Pro nabíjení akumulátoru byla zvolena U charakteristika viz. Strana 26. Po připojení akumulátoru je nabíjeno konstantním napětím o hodnotě plynovacího napětí. Proud je odebrán maximální, v ideálním případě je omezen na 0,5C a především je hlídána teplota akumulátoru. Napětí na akumulátoru se postupně zvyšuje a proud úměrně klesá, po dosažení hranice plynovacího napětí je nabíjení přepnuto na nabíjení konstantním napětím v režimu Standby, jež je určeno výrobcem. Při připojení zátěže napětí klesne pod hranici udržovacího nabíjení a režim řízení je přepnut na trackové řízení, tedy hledání maximálního výkonu solárního panelu.



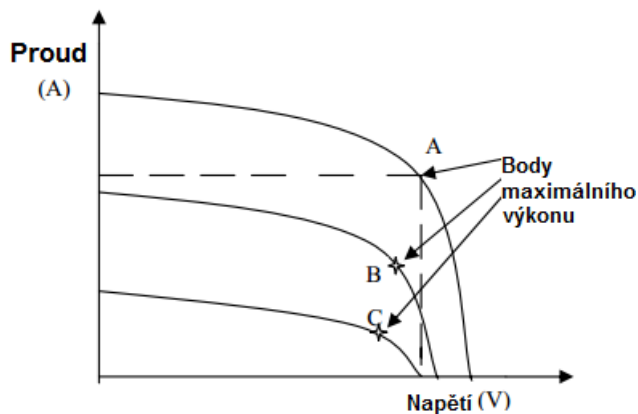
Obr.:34 Nabíjecí charakteristika VRLA akumulátoru



Obr.:35 Blokové schéma Manuálního a Automatického režimu

Trackové řízení

Jedná se o takzvané řízení se sledováním špičkového výkonu (anglicky MPPT -Maximum Power Point Tracking). Tento způsob se využívá pro zvýšení efektivnosti solárního systému.

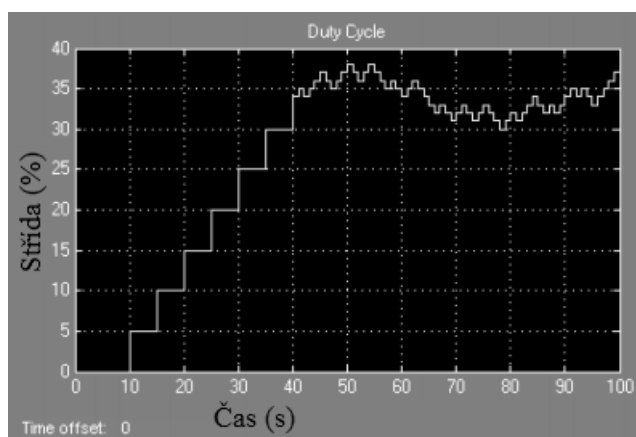


Obr.:36 Body maximálního výkonu

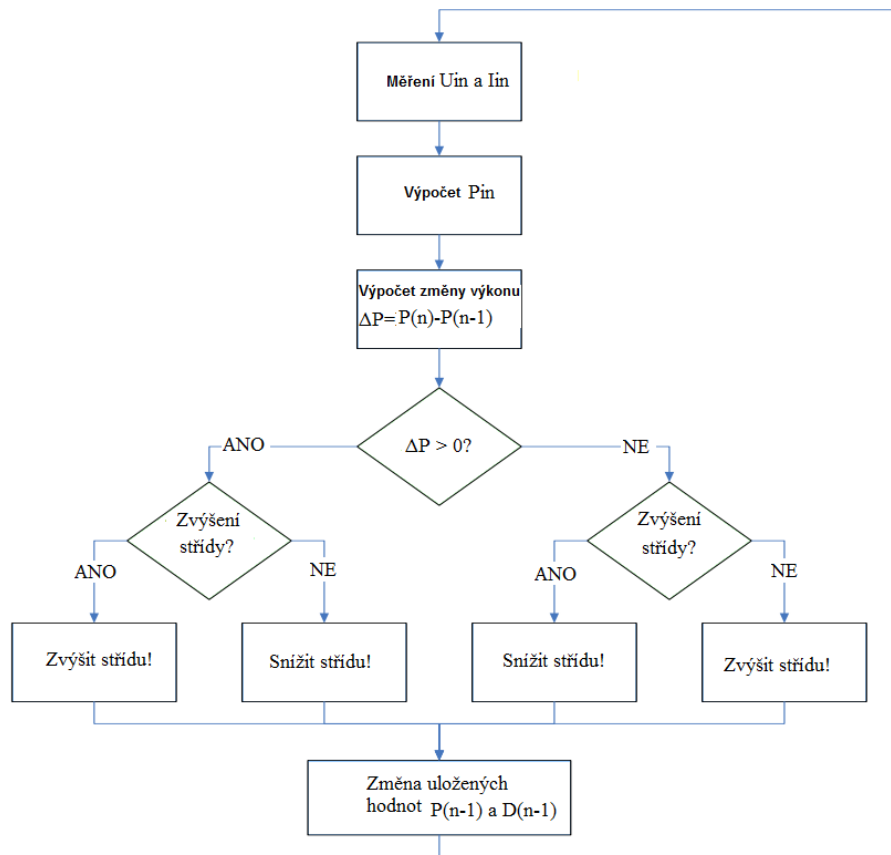
Řízení v tomto případě spočívá v měření napětí a proudu na vstupu měniče, výpočtu dodaného výkonu a na základě srovnání s předchozí hodnotou proběhne změna spínací střídy.

Změna výkonu	Změna střídy	Následující změna střídy
+	+	+
+	-	-
-	+	-
-	-	+

Tab.:2 Tabulka podmínek trackového řízení



Obr.:37 Simulace trackového řízení s pozvolným rozběhem (Soft start)[10]



Obr.:38 Vývojový diagram trackového řízení [10]

Načtení hodnot z AD převodníku (ukázka pro první 3 kanály ADC)

Přepočet pro vstupní rozsah $\pm 5V$ a odečtení offsetu, dále je nutné doplnit přepočet na reálnou hodnotu měřené veličiny dle použitého čidla / bočníku.

```

AdcData[0] = (((float32)AdcMirror.ADCRESULT0)-2055)*(-2.44140625e-3);
AdcData[1] = (((float32)AdcMirror.ADCRESULT1)-2040)*(-2.44140625e-3);
AdcData[2] = (((float32)AdcMirror.ADCRESULT2)-2054)*(-2.44140625e-3);

```

Úprava hodnot střidy PWM

Střída PWM je nastavena pro komplementární dvojici tranzistorů jedné větve "H" můstku. Rozsah nastavení je od -1 do +1 → při "-1" je plně otevřen spodní tranzistor, při hodnotě "0" je střída 50%. Pro zvyšující měnič je využíván pouze jeden tranzistor, střída je zadávána v rozsahu 0-100%

Ukázka pro nastavení prvních tří kanálů PWM

```

PWM_data[5]=(Strida_PWM*0.02-1); //Přepočet na střidu 0-100%
EPwm1Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * PWM_data[0])+PWM_MAX2);
EPwm2Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * PWM_data[1])+PWM_MAX2);
EPwm3Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * PWM_data[2])+PWM_MAX2);

```

Odeslání hodnot nastavených v LabView na DA převodník

```
SendDAC(p_AdHodnoty, AdKoeficient, AdOffsety);
```

Reset měření a potvrzení přerušení

```
AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;  
PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;  
return;
```

Vizualizace rozhraní pomocí LabView

Komunikace probíhá pomocí rozhraní USB se standardním řadičem společnosti FTDI, po připojení vývojového modulu k PC dochází dle verze operačního systému (OS) k instalaci ovladačů virtuálního USB portu. Pro běžně využívaný OS Windows XP je nutné instalovat ovladače ručně, Windows 7 a 8 instalují ovladače automaticky, avšak vyžadují tzv. elektronicky podepsané ovladače, které bohužel nejsou dostupné ani z oficiálního webu společnosti FTDI. Ovladače je možno dodatečně podepsat například pomocí utility "dseol3b", která je přiložena v datové příloze včetně ovladačů řadiče FTDI.

LabView pro komunikaci potřebuje informace o dostupných portech v PC, vyžaduje proto dodatečnou instalaci doplňku "Visa501full", dostupného na webu National Instruments, který slouží pro získávání jmen a adres prostřednictvím portů v PC. Tento doplněk je také přiložen v datové příloze.

Pro datovou komunikaci s modulem procesoru se využívá dávkový soubor automaticky generovaný CCSv4. Tento soubor obsahuje soupis všech registrů a jejich adres využitých v programu. V LabView se dále pracuje pouze se jmény těchto registrů. Při změně hodnoty registru dochází k jeho přímému přepsání v paměti, což jde využít u většiny tzv. "nechráněných" registrů, chráněné registry je nutné změnit programově a jejich změna pomocí LabView se provádí pomocí pomocných proměnných.

Pro vytvoření komunikačního rozhraní se využívá základní šablony, která obsahuje základní komunikační sekvenci, nastavení komunikace s modulem a především bloky pro posílání a čtení datových typů Unsigned Integer 16/32 a Float 32.



Obr.:39 Bloky pro čtení a zápis různých datových typů a čtení adres z dávkového souboru

Pro zadávání hodnot se využívá několik základních vizuálních prvků:

Páčkový přepínač (toggle switch)

- Typ boolean : výstupní hodnota 0 nebo 1
- Obdobou je tlačítko (push button), kontrolka (LED)

Zadávací číselné pole (numeric control)

- typ Double: výstupní hodnota 15bit plovoucí desetinná čárka

- obdobou je např. posuvný potenciometr (pointer slide), otočný potenciometr (knob)

Pro zobrazování hodnot se využívají:

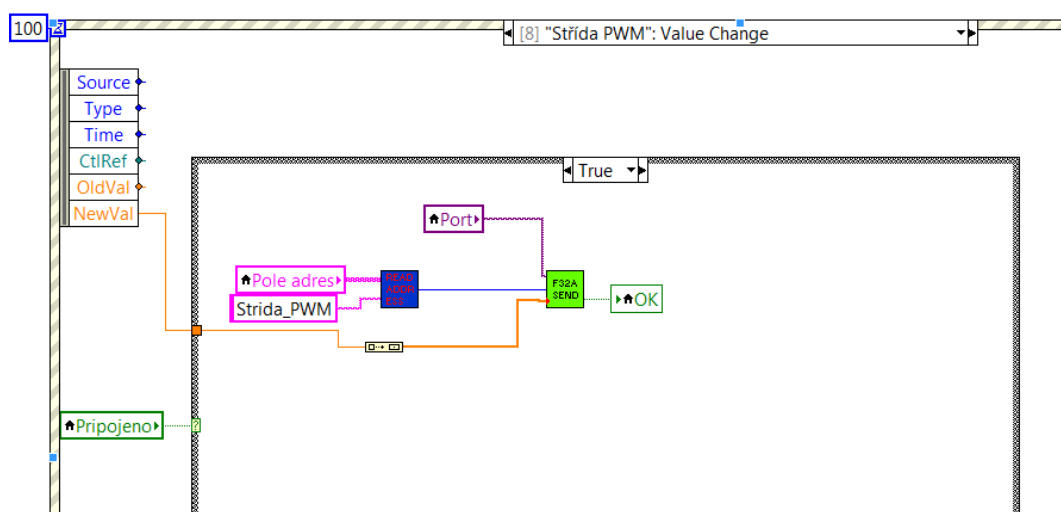
Kontrolka (LED)

- Typ boolean : výstupní hodnota 0 nebo 1

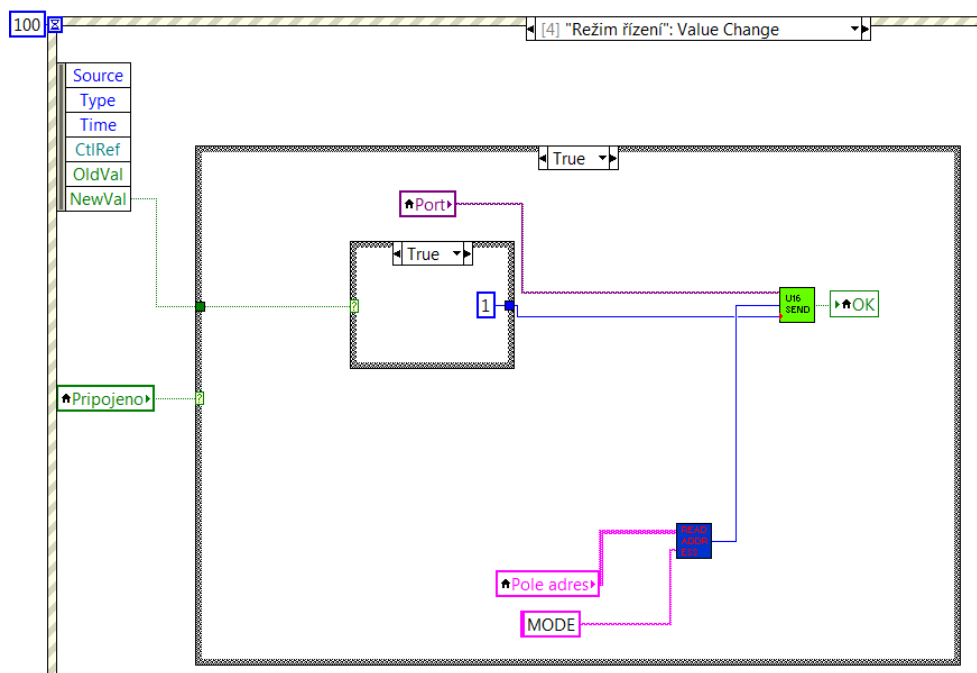
Zobrazovací číselné pole (numeric indicator)

- typ Double: výstupní hodnota 15bit plovoucí desetinná čárka
- obdobou je např. sloupcový indikátor (progress bar), ručičkový zobrazovač (meter)
Osciloskopický graf, XY graf

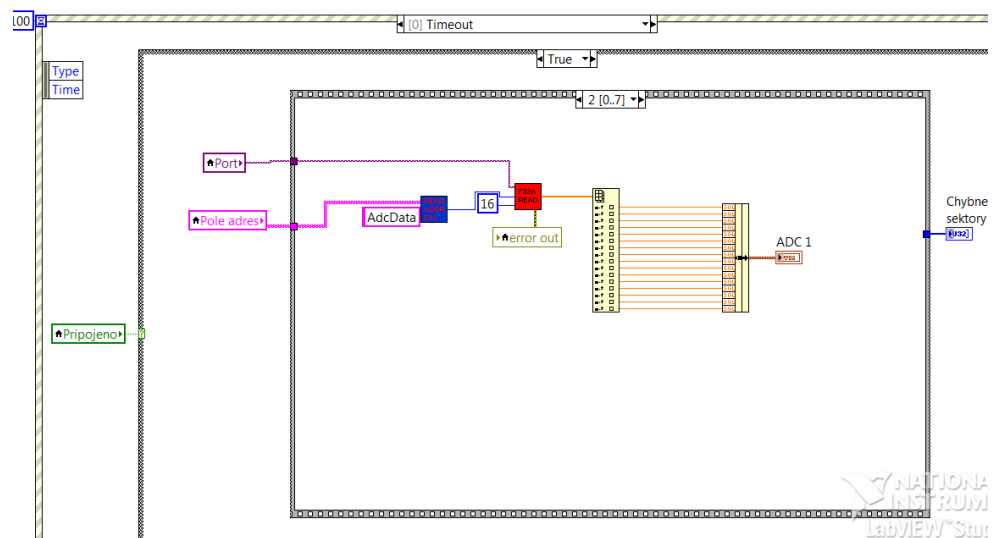
Pro příklad je zde ukázáno několik typů zadávání a čtení hodnot přes grafické rozhraní:



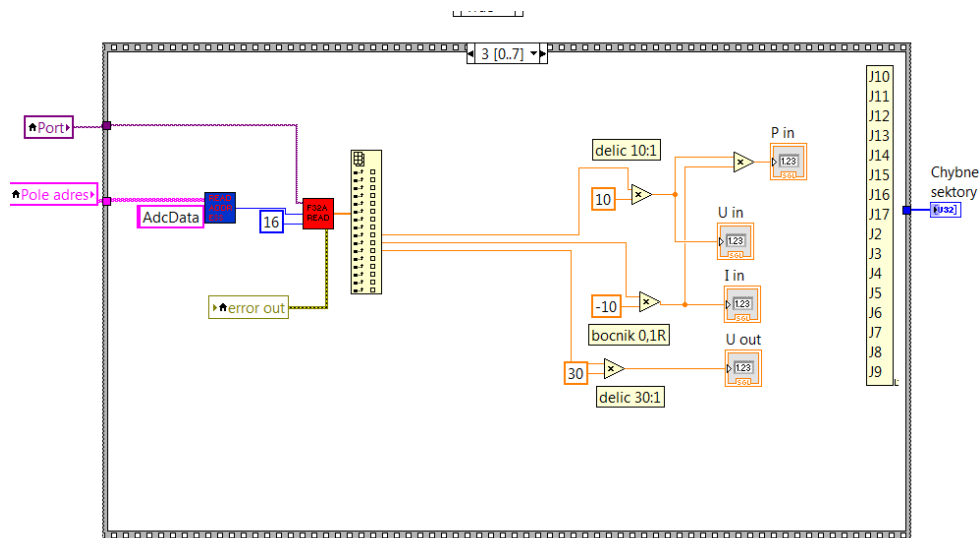
Obr.:40 Zadávání proměnné Strida_PWM typu Float 32



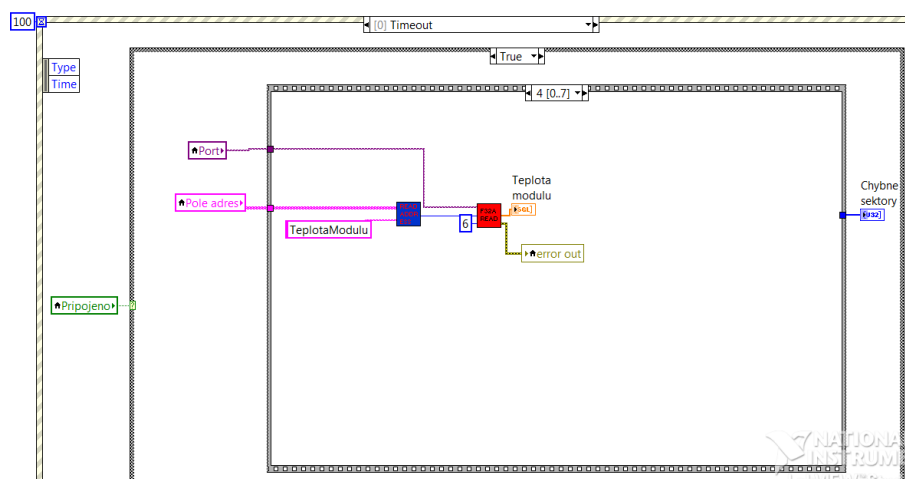
Obr.:41 Zadávání proměnné MODE typu boolean



Obr.:42 Čtení všech hodnot z AD převodníku a zobrazení pomocí osciloskopického grafu



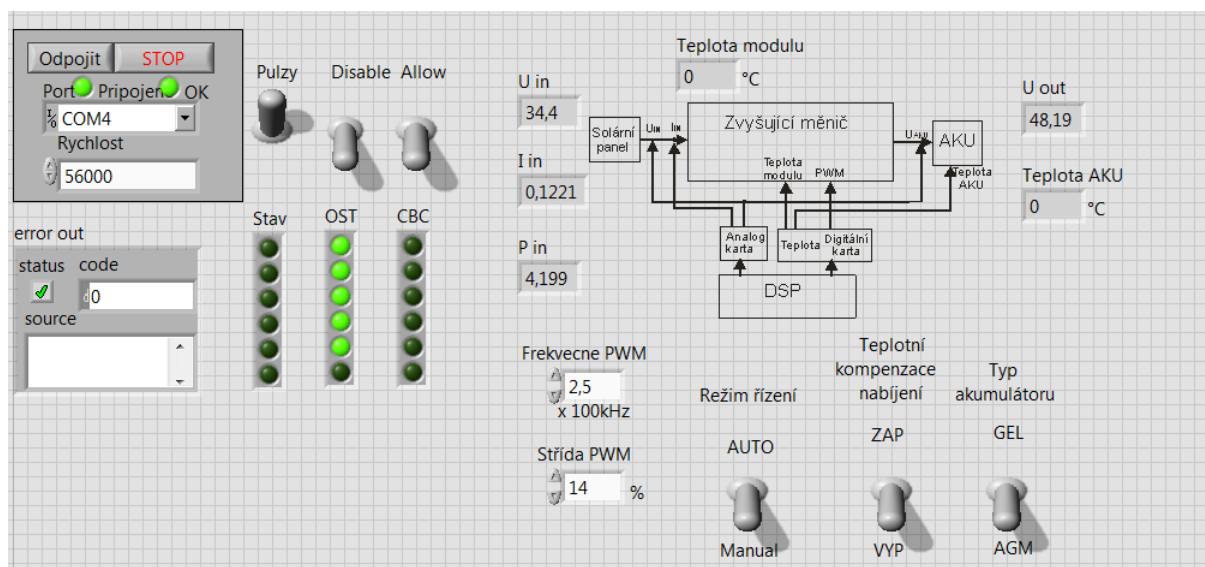
Obr.:43 Čtení vybraných hodnot z AD převodníku a zobrazení pomocí číselného pole



Obr.:44 Čtení a zobrazení hodnot teploty

Grafické rozhraní slouží jako informační a vývojový prostředek umožňující zobrazit a nastavit parametry:

- Nastavení a spuštění komunikace s modulem
- Napětí vstupů a výstupů
- Proud odebíraný ze solárních panelů
- Vstupní výkon
- Teplota spínacího prvku měniče a akumulátoru
- Výběr typu akumulátoru
- Zapnutí teplotní kompenzace nabíjení
- Zapnutí manuálního režimu řízení
 - Nastavení frekvence PWM
 - Nastavení střidy PWM



Obr.:45 Rozhraní pro nastavení parametrů

Měření:

Pro měření byly využity laboratorní přístroje:

- Osciloskop: 1. Aligent DSO1004A (50000016359/0)
2. Hameg HM1507-3 (24668/0)
Generátor: 3. Metex MXG-9810A (50000005796)
Ruční multimetr: 4. RANGE RE830B
Klešťový multimetr: 5. Pro's Kit MT-3266
Laboratorní zdroj: 6. Diametral V140R50D (1102260112)
7. Diametral L240R51D (50000024999)
8. Tesla BS554 (16024/0)

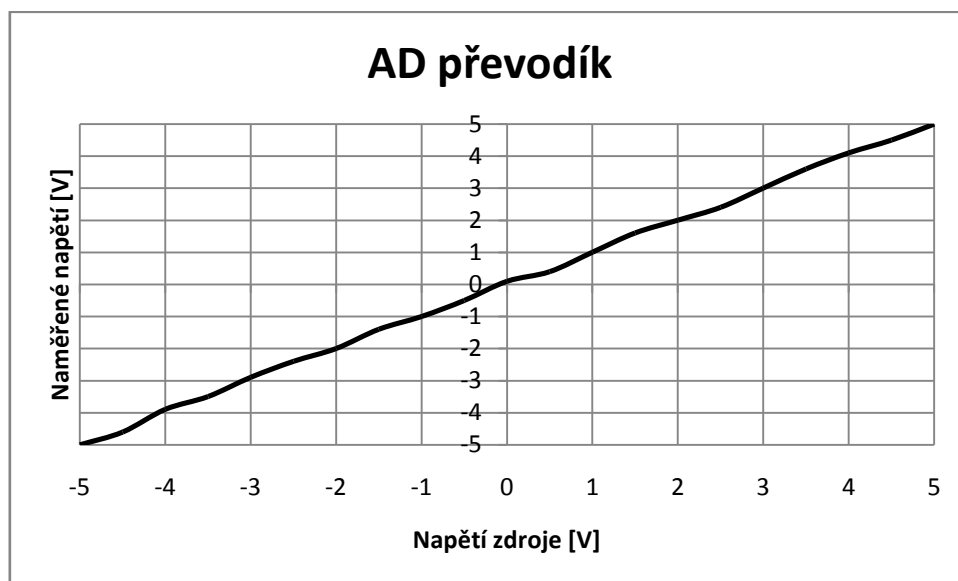


Obr.:46 Měřící stanoviště

Ověření generovaných a měřených hodnot vývojového kitu

AD převodník

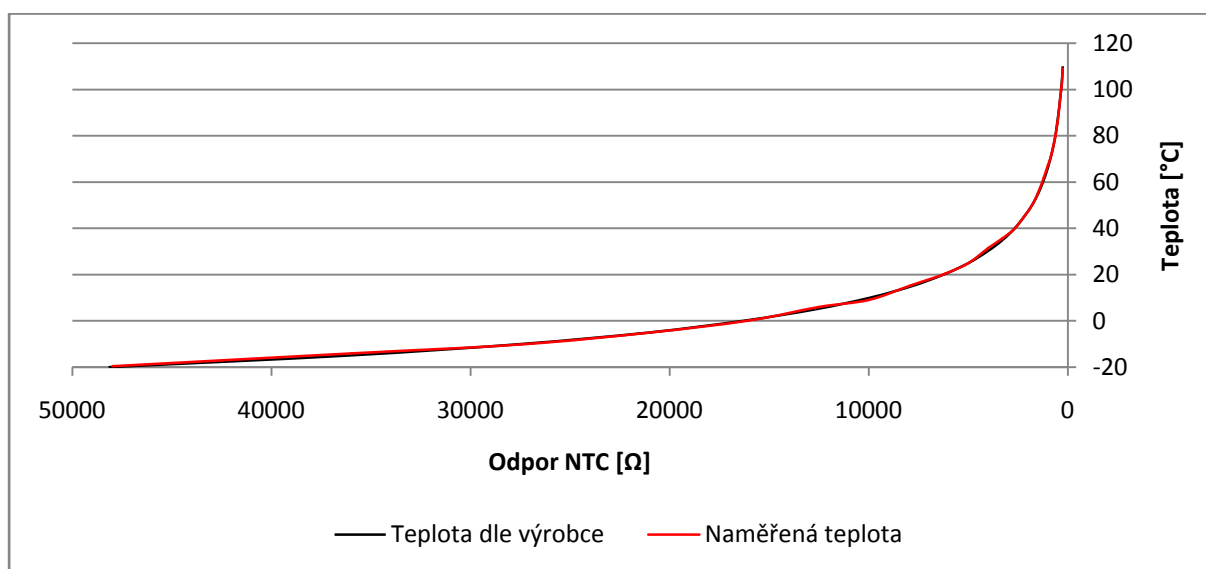
Pro generování měřeného napětí v rozsahu $U_{IN} = -5 \sim 5V$ DC byl použit regulovatelný odporový dělič tvořený více otáčkovým potenciometrem. Pro vytvoření záporné poloviny napětí byla obrácena polarita zdroje. Naměřené hodnoty byly shodné pro všechny kanály, charakteristika je tedy použitelná univerzálně pro celý AD převodník.



Graf.:1 Napětí měřené AD převodníkem

Měření teploty

Pro ověření správnosti měření teploty bylo namísto NTC čidla využito proměnného odporu s hodnotou odpovídající rozsahu NTC čidla. Tento způsob byl zvolen s ohledem na složitost měření skutečným čidlem vyžadující proměnlivý zdroj teploty, izolační komoru a teploměry. Odporová charakteristika NTC čidla je známá od výrobce, a pokud teplota udaná řídicím systémem odpovídá nastavenému odporu na vstupu, lze předpokládat správnost měření. Virtuální hodnota teploty byla odečítána z rozhraní v LabView.



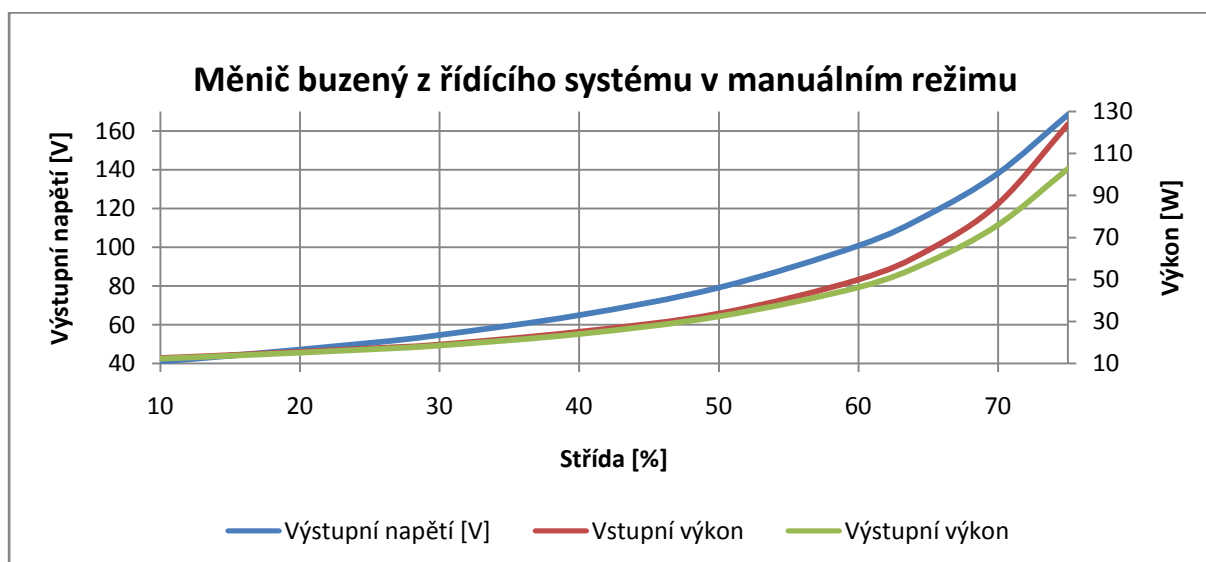
Graf.:2 Závislost zobrazované teploty na odporu NTC čidla

Měření s připojeným DC/DC měničem

Pro měření byl místo solárního panelu použit laboratorní zdroj (6.) s externím odporem, představujícím vnitřní odpor solárního panelu. Zdroj umožňuje plynulou regulaci výstupního napětí a omezení proudu, čehož bylo využito k simulaci chování solárního panelu při různém osvětlení. Jako zátěž byla využita paralelní kombinace 230V žárovek.

Měnič buzený řídicím systémem v manuálním režimu řízení

Při tomto řízení je možné ručně prostřednictvím rozhraní v LabView zadávat hodnotu frekvence a střídy PWM výstupu. Tohoto režimu řízení bylo využito pro změření zatěžovací charakteristiky použitého měniče. Měření bylo prováděno do předpokládaného maximálního výstupního napětí s ohledem na akumulátor $U_{OUT\ max} = 130V$.



Graf.:3 Měnič buzený z řídicího systému v manuálním režimu

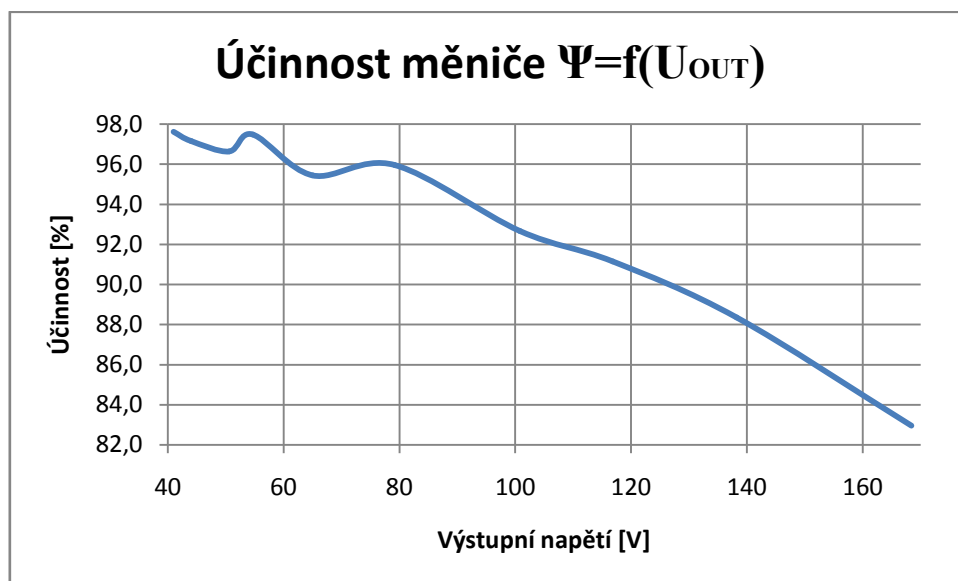
Vstupní zdroj (6.) $U_{IN}=36V$ bez proudového omezení, zátěž žárovka 230V/160W

Účinnost byla určena z naměřených vstupních a výstupních napětí a proudů. Jedná se zde především o informativní měření kvality používaného měniče.

$$P_{IN} = U_{IN} \cdot I_{IN} \quad (5)$$

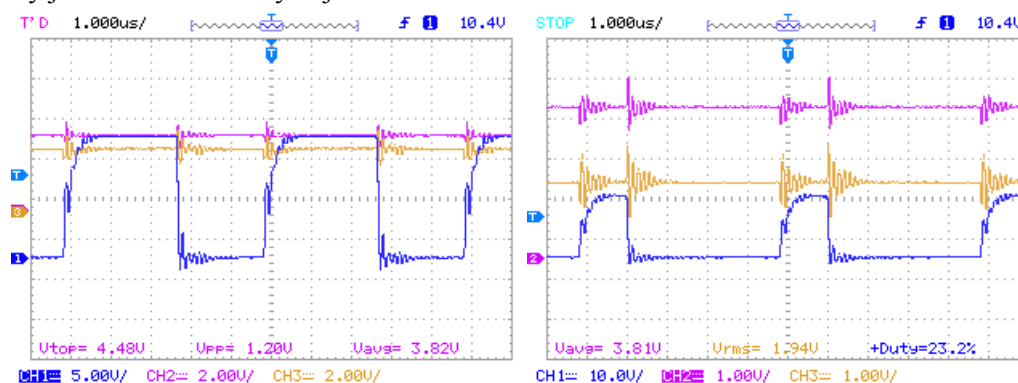
$$P_{OUT} = U_{OUT} \cdot I_{OUT} \quad (6)$$

$$\Psi = \frac{P_{OUT}}{P_{IN}} \cdot 100\% \quad (7)$$



Graf.:4 Účinnost měniče v závislosti na výstupním napětí

Pro příklad je zde doložena ukázka vzniku rušení vlivem spínání výkonového tranzistoru. Z Obr.:47 je jasné patrné, že větší rušení vzniká při vypínání tranzistoru. Toto rušení má značný vliv nejen na připojenou zátěž ale především na řídicí algoritmus využívající naměřených hodnot vstupního a výstupního napětí. Rušení se omezuje na fyzické úrovni pomocí lepší konstrukce měniče, využitím kvalitních měřicích čidel a pomocí programového řízení např. řízeným měřením v polovině PWM pulzu, kdy je zvlnění teoreticky nejmenší.



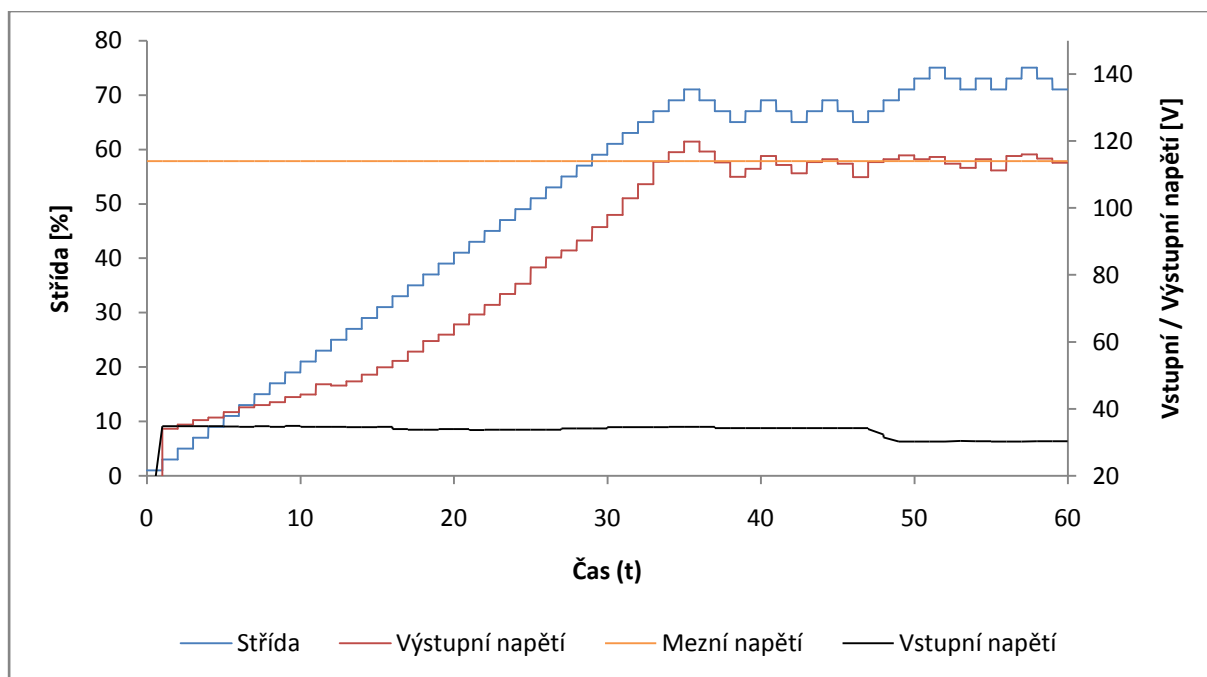
Obr.:47 Šíření rušení vzniklého spínáním výkonového tranzistoru

(modrá - PWM z budiče tranzistoru 1:1, fialová - vstupní napětí 1:10 žlutá - výstupní napětí 1:30)

Měnič buzený řídicím systémem v automatickém režimu

Při měření v automatickém režimu byly především ověřovány správné napěťové hladiny přepínací režimy nabíjení. Dalším bodem bylo vhodné nastavení časových konstant pro rozběh řízení a jednotlivé kroky při trackování. Měření automatického režimu bylo prováděno pomocí laboratorního zdroje (6.) s přidáním externím odporem $R_{IN} = 3\Omega$ simulujícím vnitřní odpor solárního panelu a jako zátěž byla využívána žárovka 230V/160W. Tyto podmínky nejsou ideální, ale dovolují ověřit funkci algoritmu při trackování. Při využití na reálném systému vybaveném solárním panelem a

akumulátorem by bylo nutné pouze upravit výše zmíněné časové konstanty dle dynamiky systému. Při měření trackového algoritmu bylo využito možnosti krokování programu v CCSv4. Tímto způsobem bylo možné dosáhnout plynulého rozběhu bez snižování střídy vlivem zákmitů a naměřeného rušení z AD převodníku. Pro omezení byla zavedena mezní hodnota napětí, protože při vyžití výše zmíněného napájecího zdroje bez akumulátoru na výstupní straně by střída stoupala až do 100% což by znamenalo trvalé sepnutí tranzistoru a vznik zkratu napájecího zdroje. Pro příklad byla změněna v průběhu hodnota vstupního napětí $U_{IN1} = 36V$ na $U_{IN2} = 30V$, z grafu je jasně patrná adaptace zvýšením střídy PWM.



Graf.:5 Měření trackového řízení

Závěr:

Cílem práce bylo vytvoření řídicího systému fotovoltaického měniče, tento cíl byl dosažen. V úvodní části je rozebrána problematika solárních systémů, jejich jednotlivých částí a způsobu spojování a případně připojení k rozvodné síti. Řídicí systém pracuje na vývojovém modulu osazeném signálovým procesorem TMS320F28335, a dovoluje tak další využití tohoto modulu i v jiných aplikacích, než pro které byl navržen. Záměrem byla realizace trackového řízení pro maximální využití energetického potenciálu fotovoltaického zdroje, toto řízení bylo vytvořeno a měřením ověřeno. Dále byl program doplněn o řízený nabíjecí cyklus v případě odpojené zátěže, teplotní ochrany akumulátoru a měniče. Posledním rozšířením byla realizace grafického rozhraní v prostředí LabView, která umožňuje vyčítání a zadávání parametrů a rozšířit tak využití také jako vývojový / výukový prostředek.

Možnost rozšíření řídicího systému:

Vytvoření "embedded" systému napájeného z měniče, odstranění nevyužívaných periférií, galvanické oddělení řídicího systému od měniče, vybavení systému měřicími čidly.

Doplnění o LCD displej a manuální ovládací panel:

Umožní bez nutnosti komunikačního rozhraní vytvořit univerzální systém s měničem pro různé typy akumulčních jednotek, rychle a přehledně zobrazí případné poruchy a chyby systému.

Doplnění o komunikační rozhraní:

Umožní uživateli přehled nad aktuálním stavem systému prostřednictvím PC, vzdálený přístup a řízení systému.

Doplnění SW vybavení:

Doplnění o další typy podporovaných akumulátorů a vytvoření jejich nabíjecích algoritmů, ochrana solárních panelů proti přehřátí. Možné rozšíření pro více měničů řízených jedním systémem.

Literatura:

- [1] Krejčířík A., DC/DC měniče. Praha, BEN 2002. ISBN 80-7300-045-8
- [2] Krejčí A., Napájecí zdroje 1. Praha, BEN 1997. ISBN 80-86056-02-3
- [3] Elektronika pro kempink a karavan. Sbírka příkladů, překlad německého originálu. Ostrava, HEL 1997. ISBN 80-902059-7-6
- [4] 305 zajímavých zapojení 2. díl. Sbírka příkladů, překlad německého originálu. Ostrava, HEL 2003. ISBN 80-902059-5-X
- [5] Bc. Chalupka Tomáš. Diplomová práce - Měnič pro fotovoltaiiku, VŠB-TUO 2014

Elektronické zdroje:

Dostupné ke dni 1. 5. 2014

- [6] <http://www.motola.cz/UserFiles/Diskuzni_clanky/akumulatory.pdf>
- [7] <<http://www.oez.cz/aktuality/obecne-schema-zapojeni-fotovoltaickeho-zdroje>>
- [8] <<http://www.cez.cz/edee/content/microsites/solarni/k41.htm>>
- [9] <<http://www.cez.cz/edee/content/microsites/solarni/k32.htm>>
- [10] <http://cegt201.bradley.edu/projects/proj2012/pvc/files/PVC_final_report.pdf>
- [11] <<http://cdn.intechopen.com/pdfs-wm/12693.pdf>>
- [12] <http://cs.wikipedia.org/wiki/Fotovoltaick%C3%BD_%C4%8Dl%C3%A1nek>
- [13] <<http://cz.farnell.com/fw-bell/na25/current-sensor-hall-effect-25a/dp/2292986>>
- [14] <http://danyk.cz/menic230_6.html>
- [15] < <http://www.pwr.com/pwr/docs/cm200dx-24s.pdf>>
- [16] <<http://eshop.neosolar.cz/userfiles/file/v%C3%BDb%C4%9Br%20baterie%20pro%20ostrovn%C3%AD%20syst%C3%A9my.pdf>>

Přílohy

Seznam tištěných příloh:

- Program C++ II

Seznam elektronických příloh (přiloženo na CD/DVD)

- Program "dseo13b.exe"
- Doplněk k LabView pro komunikaci přes USB
- Kompletní projekt s programem v CCSv4
- Uživatelské prostředí pro LabView

Program C++

V programu jsou smazány nevyužívané registry např. od AD převodníku a delší bloky komentářů. Všechny jsou dostupné v plné verzi programu na doprovodném DVD, popř. v elektronické příloze této práce.

```
/*
*****
main.c hlavni zdrojovy soubor
DP STRELEC MARTIN
RIDICI SYSTEM PRO FOTOVOLTAICKY MENIC
*****
#include "DSP28x_Project.h"
#include "ENET.h"
#include "regmacro.h"
#include "math.h"

//definice promennych
Uint16 MODE = 0;      //prepinani manualni a automaticky rezim 0 rucni, 1 auto
float32 Strida_PWM = 50;      //nastaveni stridy automatika
float32 Strida_PWM_lab = 50; //nastaveni stridy z labview
float32 PWM_lab = 375;      //nastaveni frekvence z labview
Uint16 AKU = 1;      //nastaveni typu AKU (1=AGM, 0=GEL)
Uint16 teplota_mode = 0; //zapnutí teplotní kompenzace (0=vyp, 1=zap)
Uint16 pocet_AKU = 8;      //počet 12V AKU
Uint16 a = 0;      //promenne pro pocitadla
Uint16 b = 0;      //promenne pro pocitadla
Uint16 x = 0;      //promenne pro pocitadla
Uint16 rozbeh = 0;      //rozbeh pri automatickem rizeni
float32 proudove_omezeni = 5; //velikost proudoveho omezeni pri rozbehu

//konstanty pro pocatecni nastaveni PWM
float32 PWM_MAX=750;      //frekvence PWM (75MHz/PWM_MAX=freq PWM)

float32 PWM_MAX2=375;      //pomocna hodnota pro určení frekvence
PWM_MAX2=PWM_MAX/2
float32 PWM_PRC=7.5;      // PWM_PRC=PWM_MAX/100
float32 PWM_FED=0;      // nastaveni ostatnich parametru PWM
float32 PWM_RED=0;
float32 PWM_DIV=0;
float32 PWM_HSD=0;
```

```

// Deklarace funkci v tomto souboru
interrupt void adc_isr(void);    //preruseni od AD prevodniku

// cteni hodnot
float32 Uvstup = 0;              //vstupni napeti
float32 Uvystup = 0;             //vystupni napeti
float32 Proud = 0;               //vstupni proud
float32 P_vstup = 0;             //vstupni prikon
float32 P_stare = 0;             //prikon pro trackovani
float32 P_nove = 0;              //prikon pro trackovani
float32 U1 = 0;                  //vstupni Napeti pro trackovani
float32 I1 = 0;                  //Proud pri trackovani
float32 U2 = 0;                  //vystupni napeti pri trackovani
Uint16 Nabijeni = 0;             //mod nabijeni (0=nabijeni, 1=standby 2 track)
float32 U_max = 0;               //Maximalni napeti aku
float32 U_stb = 0;               //napeti pro standby rezim
float32 U_zatez = 0;             //hranice pripojeni zeteze
float32 deltat = 0;              //zmena teploty
float32 kompenzace = 0;           //hodnota teplotni kompenzace
float32 strida_stara = 0;         // hodnota stridy pri trackovani
float32 strida_nova = 1;         // hodnota stridy pri trackovani
float32 strida_zmena = 1;        //zmena stridy pri trackovani
float32 P_zmena = 0;             //zmena prikovu pri trackovani
float32 zmena = 0;               //promenna zmeny stridy pri trackovani

//promenne pro komunikaci v LabView
Uint16 pole_scia[256];
Uint16 uk_pole_scia = 0x1234;
Uint16 i = 0; //vseobecne pocitadlo pro main
Uint16 znak_scia = 0;
Uint16 status_scia = 0;
Uint32 timeout_scia = 0;
Uint16 *p_adresa_scia = 0;
Uint32 adresa_scia = 0;
Uint16 crc1_scia = 0xFFEE;
Uint16 crc2_scia = 0xFFEE;
Uint16 data = 0;
Uint32 k_data = 0;
Uint32 neco = 0;
Uint32 nic = 0;

```

```

Uint16 nacti = 0;
Uint16 uloz = 0;
Uint16 EepromAdresa = 0;
float32 ppp=5;

/* Komunikace SpiC*/
Uint16 SpiCSendData = 0;
Uint16 SpiCReadData = 0;
Uint16 SpiCStatus = 0;
Uint16 SpiCCount = 0;
Uint16 SpiCData[36];
Uint16 SpiCPlatnaData = 1; //1-neplatne 0-platne
Uint16 SpiCcrc = 0x0000;
float32 TeplotaModulu[6] = {0.0,0.0,0.0,0.0,0.0,0.0};
float32 TeplotaMax = 110;

float32 AdcData[16] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

//Pocatecni hodnota PWM
float32 PWM_data[6] = {0.5, 0.5, 0.5, 0.5, 0.5, 0.5};

ABC Umot = {0,0,0};
ABC abc_sPWM = {0,0,0};
SSS sss_sPWM = {0,0};
float32 wt = 0;
float32 f = 1.0; //Frekvence v Hz

#define PI 3.141592654
#define PI2 6.283185307
#define PI23 2.094395102
#define deltaUt0.24 //teplotni kompenzace pro 96V aku (0,03V/1°C pro
12V aku)vyssi teplota = nizsi napeti

//data pro DA převodník
float32 *p_AdHodnoty[4] = {&Umot.A, &Umot.B, &Umot.C, &Umot.A}; //Pocatecni nastaveni
DA vystupu
float32 AdKoeficient[4] = {1.0,1.0,1.0,1.0}; //Pocatecni nastaveni prevodu ADC pro
Labview
float32 AdOffsety[4] = {0.0,0.0,0.0,0.0};

```

```

//PWMovladani pulzu
Uint16 PWMon = 0x0000;      //xxxx xxxx xx65 4321
Uint16 PWMoff = 0x0000;     //xxxx xxxx xx65 4321

void main(void)
{
// Inicializace systemovych soucasti
// PLL, WatchDog, Peripheral Clocks, atd.
    InitSysCtrl();

// inicializace GPIO:
    EALLOW;
    GpioDataRegs.GPASET.bit.GPIO30 = 1;
    GpioDataRegs.GPBSET.bit.GPIO34 = 1;
    GpioCtrlRegs.GPAMUX2.bit.GPIO30 = 0;
    GpioCtrlRegs.GPADIR.bit.GPIO30 = 1;//RELE1
    GpioCtrlRegs.GPBMUX1.bit.GPIO34 = 0;
    GpioCtrlRegs.GPBDIR.bit.GPIO34 = 1;//RELE2
    GpioCtrlRegs.GPBMUX1.bit.GPIO32 = 2;//PWMSYNCIN
    GpioCtrlRegs.GPBQSEL1.bit.GPIO32 = 2;
    GpioCtrlRegs.GPBDIR.bit.GPIO32 = 0;
    GpioCtrlRegs.GPBMUX1.bit.GPIO33 = 2;//PWMSYNCCOUT
    GpioCtrlRegs.GPBDIR.bit.GPIO33 = 1;
    EDIS;

// Zakazani vseh prerusnei CPU
    DINT;

// Inicializace PIE registru na vychodi stav.
    InitPieCtrl();

// Zakazani prerusni CPU a vynulovani vseh priznaku:
    IER = 0x0000;
    IFR = 0x0000;

// Inicializace vseh PIE vektoru preruseni na ukazatele ISR
    InitPieVectTable();

// Premapovani vektoru prerusni se provede timto zapisem:
    EALLOW; // povoleni zapisu do chraneneho registru
    PieVectTable.ADCINT = &adc_isr;

```

```

EDIS; // ukončení zápisu do chráněných registrů
// Inicializace Periferií.
InitGpio();
InitSpiB();
InitSpiBGpio();
InitSpiC();
InitSpiCGpio();
InitScia();
InitScib();
InitAdc();
ConfigAdc();
InitExRAM();

//Inicializace PWM + nastavení počátečních hodnot 0
InitEPwm(PWM_MAX, PWM_HSD, PWM_DIV, PWM_FED, PWM_RED);
EPwm6Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * 0)+PWM_MAX2);
// povolení prerusení:
IER |= M_INT1; //povolení proerusnei ve skupine 1 (AD převodník)
// Povolení prerusnei od ADC v PIE skupina 1 preruseni 6
PieCtrlRegs.PIEIER1.bit.INTx6 = 1;
// Povolení všech prerusení:
EINT; // Povolení všech preruseni INTM
ERTM; // povolení preruseni realtime beduggeru DBGEM
/*inicializace DA převodníku*/
McbSpaRegs.DXR2.all=0x0028;
McbSpaRegs.DXR1.all=0x0001;
while(McbSpaRegs.SPCR2.bit.XRDY == 0){
for(i = 0; i<256; i++){
    pole_scia[i] = 0x00FF;
}
init_crc16_tab();
neco = 0x12345678;
GpioDataRegs.GPACLEAR.bit.GPIO30 = 1;
//Nekonečná smyčka: (Komunikace s LabView)
while(1)
{
    if(SciaRegs.SCIFFRX.bit.RXFFST > 0){
        znak_scia = SciaRegs.SCIRXBUF.all; //Čekání na znak
        //Vetvení podle pořadí znaku v řetězci
        switch(status_scia){
            case 0:{//čtení dat

```



```

        if(znak_scia == 0x0055 || znak_scia == 0x00AA){
            status_scia = 1;
            uk_pole_scia = 1;
            pole_scia[0] = znak_scia;
            timeout_scia = 0;
        }
        break;
    }
    case 1:{
        pole_scia[uk_pole_scia] = znak_scia;
        status_scia = 2;
        timeout_scia = 0;
        break;
    }
    case 2:{
        uk_pole_scia++;
        pole_scia[uk_pole_scia] = znak_scia;

        if((uk_pole_scia + 1) == pole_scia[1]){
            status_scia = 3;
            k_data++;
        }
        break;
    }
    default:{
        status_scia = 0;
    }
}

}
//Timeout pro komunikaci s Labview
else{
    timeout_scia++;
    if(timeout_scia > 200000000){
        status_scia = 0;
        timeout_scia = 0;
    }
}

if(status_scia == 3){//kontrola a dekodovani prijate zpravy
    crc1_scia = 0x0000;
    for(uk_pole_scia = 0; uk_pole_scia < pole_scia[1]; uk_pole_scia++){
        crc1_scia = update_crc_16(crc1_scia, pole_scia[uk_pole_scia]);
    }
}

```

```

    }
    if(crc1_scia == 0x0000){
        status_scia = 4;
        timeout_scia = 0;
        adresa_scia = 0x000000FF & (UInt32)pole_scia[5];
        adresa_scia = adresa_scia + (0x0000FF00 & ((UInt32)pole_scia[4]<<8));
        adresa_scia = adresa_scia + (0x00FF0000 & ((UInt32)pole_scia[3]<<16));
        adresa_scia = adresa_scia + (0xFF000000 & ((UInt32)pole_scia[2]<<24));
        p_adresa_scia = (UInt16*)adresa_scia;
        if(pole_scia[0] == 0x0055){
            for(uk_pole_scia = 0; uk_pole_scia < pole_scia[6]; uk_pole_scia++){
                data = p_adresa_scia[uk_pole_scia];
                pole_scia[(uk_pole_scia<<1)+7] = data & 0x00FF;
                pole_scia[(uk_pole_scia<<1)+8] = (data & 0xFF00)>>8;
            }
            pole_scia[1] = (uk_pole_scia<<1)+9;
        }
        else{
            for(uk_pole_scia = 0; uk_pole_scia < pole_scia[6]; uk_pole_scia++){
                data = (0x00FF & pole_scia[(uk_pole_scia<<1) + 8])<<8;//MSB
                data += 0x00FF & pole_scia[(uk_pole_scia<<1)+7];//LSB
                p_adresa_scia[uk_pole_scia] = data;
            }
            pole_scia[1] = 8;
        }
        crc2_scia = 0x0000;
        for(uk_pole_scia = 0; uk_pole_scia < pole_scia[1]-2; uk_pole_scia++){
            crc2_scia = update_crc_16(crc2_scia, pole_scia[uk_pole_scia]);
        }
        pole_scia[pole_scia[1]-2] = (0x00FF & crc2_scia);
        pole_scia[pole_scia[1]-1] = (0xFF00 & crc2_scia)>>8;
        uk_pole_scia = 0;
    }
}
if(status_scia == 4){
    /*realizace odesilani dat*/
    timeout_scia = 0;
    if(uk_pole_scia != pole_scia[1]){
        if(SciaRegs.SCIFFTX.bit.TXFFST < 16){
            SciaRegs.SCITXBUF = pole_scia[uk_pole_scia];
            uk_pole_scia++;
        }
    }
}

```

```

        }
    }
    else{
        status_scia = 0;
    }
}/*konec komunikace labview*/
//povolovani pulzu
    if((PWMon & 0x0020) > 0){
        PWMon -= 0x0020;
        EALLOW;
        EPwm6Regs.TZCLR.bit.OST = 1;
        EPwm6Regs.TZCLR.bit.CBC = 1;
        EDIS;
    }
//konec povolovani pulzu
    if((PWMoff & 0x0020) > 0){
        PWMoff -= 0x0020;
        EALLOW;
        EPwm6Regs.TZFRC.bit.OST = 1;
        EDIS;
    }
//konec zakazani pulzu
    if(ScibRegs.SCIFFRX.bit.RXFFST > 0){
        ScibRegs.SCITXBUF = ScibRegs.SCIRXBUF.bit.RXDT;
    }
    /*realizace cteni teploty (Komunikace s ATmega )
    if(SpiCStatus == 0){
//test statusu prenosu sci tento blok realizuje start komunikace
        GpioDataRegs.GPBCLEAR.bit.GPIO60 = 1;//CS1 - ATmega168
        McbspbRegs.DXR1.all=0x55;
        SpiCPlatnaData = 1;
        SpiCStatus = 1;//predava kontrolu na dalsi sekci
    }
    if(McbspbRegs.SPCR1.bit.RRDY == 1 && SpiCStatus == 1 ) { //kontrola
konce prenosu spi
        GpioDataRegs.GPBSET.bit.GPIO60 = 1;//CS1 - uvolneni CS
        SpiCReadData = McbspbRegs.DRR1.all;//prazdne cteni
        SpiCStatus = 2;//predani rizeni dale
    }
    if(SpiCStatus == 2){ //realizace prazdnych zapisu pro cteni z ATmega168
        GpioDataRegs.GPBCLEAR.bit.GPIO60 = 1;//CS1

```

```

        McbspbRegs.DXR1.all=0x00;//prazdne data
        SpiCStatus = 3;//predani rizeni dale
    }
    if(McbspbRegs.SPCR1.bit.RRDY == 1 && SpiCStatus == 3) { //kontrola konce
prijmu spi
        GpioDataRegs.GPBSET.bit.GPIO60 = 1;//CS1
        SpiCReadData = McbspbRegs.DRR1.all;
        if(SpiCPlatnaData > 0){
            if(SpiCReadData == 0x55){
                SpiCPlatnaData = 0;
                neco = 0;
            }
            else
                {neco++;}
        }
        else{
            SpiCData[SpiCCount] = SpiCReadData;//ulozeni prijateho
bytu
            SpiCCount++; //zvyseni ukazatele
        }
        SpiCStatus = 2;//predani rizeni vyse
    }
    if(SpiCCount > 30){ //po prijati 28 bytu z ATmega168 predvod na float32
        SpiCcrc = 0x0000;
        for(SpiCCount = 0; SpiCCount < 30; SpiCCount++){
            SpiCcrc = update_crc_16(SpiCcrc, SpiCData[SpiCCount]);
        }
        SpiCStatus = 0;//resetovani prenosu SPI
        SpiCCount = 0;//nulovani pocitadla prijatych bytu
        if(SpiCcrc == 0){
            TeplotaModulu[0] = Array2Float(SpiCData);
            TeplotaModulu[1] = Array2Float(SpiCData+4);
            TeplotaModulu[2] = Array2Float(SpiCData+8);
            TeplotaModulu[3] = Array2Float(SpiCData+12);
            TeplotaModulu[4] = Array2Float(SpiCData+16);
            TeplotaModulu[5] = Array2Float(SpiCData+20);
            PrutokVody = Array2Float(SpiCData+24);
        }
        else{
            nic++;
        }
    }

```

```

    }
}
//Preruseni od ADC - konec prevodu
interrupt void adc_isr(void)
{
    //nastaveni typu akumulátoru z labview
    if (AKU==1) //aku AGM
    {
        U_max= pocet_AKU * 14.3 + kompenzace;
        U_max2=pocet_AKU * 14.4 + kompenzace;
        U_stb= pocet_AKU * 13.8 + kompenzace;
        U_zatez= pocet_AKU * 13.6 + kompenzace;
    }
    if (AKU==0)//aku gel
    {
        U_max= pocet_AKU * 14.0 + kompenzace;
        U_max2= pocet_AKU * 14.1 + kompenzace;
        U_stb= pocet_AKU * 13.6 + kompenzace;
        U_zatez= pocet_AKU * 13.4 + kompenzace;
    }
    if (teplota_mode==1) //vypocet teplotni kompenzace
    {
        deltat=TeplotaModulu[0]-25;
        //vypocet odchylky od normalni teploty 25°C
        if(deltat > 0)//teplotni kompenzace pro vyssi teplotu
            {kompenzace = -deltat * deltaUt; }
        else //teplotni kompenzace pro nizsi teplotu
            {kompenzace = -deltat * deltaUt; }
    }
    else
        {kompenzace = 0; }
    if (MODE==1) //pokud je zapnut automaticky rezim
    {
        if (PWM_MAX != 375) //nastaveni na 200kHz
        {
            PWM_MAX = 375; //Nastaveni promenne PWM_MAX na 200kHz
            PWM_MAX2 = PWM_MAX/2; //Vypocet PWM_MAX2
            PWM_PRC = PWM_MAX/100; //Vypocet PWM_PRC
            InitEPwm(PWM_MAX, PWM_HSD, PWM_DIV, PWM_FED, PWM_RED);//Inicializace PWM

```

```

EPwm6Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * 0)+PWM_MAX2); //Zmena na
nastavenou hodnotu
        EALLOW;                //Povoleni zapisi do registru
        EPwm6Regs.TZFRC.bit.OST = 1; //zakazani pulzu
        EPwm6Regs.TZCLR.bit.OST = 1; //povoleni pulzu
        EPwm6Regs.TZCLR.bit.CBC = 1; //povoleni pulzu
        EDIS;                //Zakazani zapisu do registru
    }
    if (rozbeh == 0 && Nabijeni==0) //prvotni rozbeh na 60%
    {
        AdcData[9] = (((float32)AdcMirror.ADCRESULT9)-2048)*(-2.44140625e-3);
        Proud = AdcData[9] * -10; //prepocet proud, 0-10A, bocnik 10:1
        if (proudove_omezeni > Proud) //omezeni rozbehoveho proudu
        {
            a = a + 1;
            Strida_PWM=a; //postupne zvyseni stridy
            PWM_data[5]=(Strida_PWM*0.02-1); //prepocet na 0-100% strida
            EPwm6Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * PWM_data[5])+PWM_MAX2);
            for(x=0 ; x<50000 ; x++); //casove zpozdeni pro ustaleni
            if(a == 60)
            {
                a = 0;
                rozbeh = 1;
            }
        }
    }
    else
    {
        if (Nabijeni == 0) //U_max pokud je zapnuto nabijeni
        {
            AdcData[8] = (((float32)AdcMirror.ADCRESULT8)-2048)*(-2.44140625e-3);
            AdcData[9] = (((float32)AdcMirror.ADCRESULT9)-2048)*(-2.44140625e-3);
            AdcData[10] = (((float32)AdcMirror.ADCRESULT10)-2048)*(-2.44140625e-3);
            Uvstup = AdcData[8] * 10; //prepocet napeti 0-50V delic 10:1
            Proud = AdcData[9] * -10; //prepocet proud, 0-10A, bocnik 10:1
            Uvystup = AdcData[10] * 30; //prepocet, 0-150V, delic 30:1
            if(Uvystup < U_max) //U_max nastaveni stabilniho vystupniho napeti
            {
                Strida_PWM=Strida_PWM + 1;
                PWM_data[5]=(Strida_PWM*0.02-1); //prepocet na 0-100% strida
            }
        }
    }
}

```

```

EPwm6Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * PWM_data[5])+PWM_MAX2);
//nastaveni stridy na PWM
}
else
{
Strida_PWM=Strida_PWM - 1;
PWM_data[5]=(Strida_PWM*0.02-1); //prepocet na 0-100% strida
EPwm6Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * PWM_data[5])+PWM_MAX2);
}
if(Uvystup > U_max)//po dosažení nabití přepnout do režimu standby
{Nabijeni = 1;//prechod na rezim standby }
}
if (Nabijeni == 1) //rezim standby - udrzovací nabijeni
{
//vycteni analogovych hodnot
AdcData[8] = (((float32)AdcMirror.ADCRESULT8)-2048)*(-2.44140625e-3);
AdcData[9] = (((float32)AdcMirror.ADCRESULT9)-2048)*(-2.44140625e-3);
AdcData[10] = (((float32)AdcMirror.ADCRESULT10)-2048)*(-2.44140625e-3);
Uvstup = AdcData[8] * 10;//prepocet rozsahu 0-50V delic 10:1
Proud = AdcData[9] * -10;//prepocet proud,0-10A, bocnik 10:1
Uvystup = AdcData[10] * 30;//prepocet 0-150V, delic 30:1
if(Uvystup < U_stb) //U_stb
{
Strida_PWM=Strida_PWM + 1;
PWM_data[5]=(Strida_PWM*0.02-1); //prepocet na 0-100% strida
EPwm6Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * PWM_data[5])+PWM_MAX2);
}
else
{
Strida_PWM=Strida_PWM - 1;
PWM_data[5]=(Strida_PWM*0.02-1); //prepocet na 0-100% strida
EPwm6Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * PWM_data[5])+PWM_MAX2);
}
}
}
}
if (Uvystup < U_zatez )//pokud je pripojena zatez
{Nabijeni = 2;
b = b + 1;
if(b == 5000)
{
b = 0;

```

```

AdcData[8] = (((float32)AdcMirror.ADCRESULT8)-2048)*(-2.44140625e-3);
AdcData[9] = (((float32)AdcMirror.ADCRESULT9)-2048)*(-2.44140625e-3);
//nacteni potrebných hodnot z AD
    U1 = AdcData[8] * 10; //prepocet rozsahu 0-50V delic 10:1
    I1 = AdcData[9] * -10; //prepocet proud, 0-10A, bocnik 10:1
    P_nove = U1 * I1;
    P_zmena = P_nove - P_stare;
        if(P_zmena > 0)
            {for(x=0 ; x<5000 ; x++);}
            strida_zmena = strida_nova - strida_stara;
if(P_zmena > 1.5 && strida_zmena > 0 && strida_nova < 98)
    {zmena = 2;}
if(P_zmena > 1.5 && strida_zmena < 0 && strida_nova > 2)
    {zmena = -2;}
if(P_zmena < -1.5 && strida_zmena > 0 && strida_nova > 2)
    {zmena = -2;}
if(P_zmena < -1.5 && strida_zmena < 0 && strida_nova < 98)
    {zmena = 2;}
    strida_stara = strida_nova;
    strida_nova = strida_nova + zmena;
    P_stare = P_nove;
    PWM_data[5] = (strida_nova * 0.02 - 1);
    EPwm6Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 *
PWM_data[5]) + PWM_MAX2); //nastaveni stridy na PWM
    }
}
else
    {Nabijeni = 0; //pokud je zatez odpojena
}

}
if (MODE == 0) //pokud je zapnut manualni rezim
{
    if (Strida_PWM != Strida_PWM_lab)
    {
        Strida_PWM = Strida_PWM_lab;
        PWM_data[5] = (Strida_PWM * 0.02 - 1); //prepocet na 0-100%
        EPwm6Regs.CMPA.half.CMPA =
(Uint16)((PWM_MAX2 * PWM_data[5]) + PWM_MAX2); //nastaveni stridy na PWM
    }
    if (PWM_MAX != PWM_lab) // pri zmene hodnoty frekvence z LabView

```



```

{
    PWM_MAX=PWM_lab; //Nastaveni promenne PWM_MAX na zmenenou
    PWM_MAX2=PWM_MAX/2; //Vypocet PWM_MAX2
    PWM_PRC=PWM_MAX/100; //Vypocet PWM_PRC
    InitEPwm(PWM_MAX, PWM_HSD, PWM_DIV, PWM_FED, PWM_RED)

    EPwm6Regs.CMPA.half.CMPA = (Uint16)((PWM_MAX2 * 0)+PWM_MAX2);
    EALLOW; //Povoleni zapisi do registru
        EPwm6Regs.TZFRC.bit.OST = 1; //zakazani pulzu
        EPwm6Regs.TZCLR.bit.OST = 1; //povoleni pulzu
        EPwm6Regs.TZCLR.bit.CBC = 1; //povoleni pulzu
    EDIS; //Zakazani zapisu do registru
}
}
/*vyceti analogovych hodnot
AdcData[8] = (((float32)AdcMirror.ADCRESULT8)-2048)*(-2.44140625e-3);
AdcData[9] = (((float32)AdcMirror.ADCRESULT9)-2048)*(-2.44140625e-3);
AdcData[10]=(((float32)AdcMirror.ADCRESULT10)-2048)*(-2.44140625e-3);
Uvstup = AdcData[8] * 10; //prepocet rozsahu 0-50V delic 10:1
Proud = AdcData[9] * -10; //prepocet proud,0-10A, bocnik 10:1
Uvystup = AdcData[10] * 30; //prepocet vystupni

SVPWM(sss_sPWM, abc_sPWM); //vypocet vektorove PWM

/*Poslani hodnot zvolenych v Labview na DA prevodniky*/
SendDAC(p_AdHodnoty, AdKoeficent, AdOffsety);

/*Reset merici sekvence*/
AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;
/*Potvrzeni preruseni od skupiny 1*/
PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
return;
} //end

```